

Instituto Superior de Engenharia do Porto

**SISTEMA DE ANIMAÇÃO DE PERSONAGENS VIRTUAIS
PARA COMUNICAÇÃO NÃO VERBAL**

Rui Luís Correia Dias

Dissertação para obtenção do Grau de Mestre em

Engenharia Informática

Área de Especialização em

Sistemas Gráficos e Multimédia

Orientador: Prof. Doutor João Paulo Jorge Pereira

Co-orientador: Prof. Doutor Ido Aharon Iurgel

Júri:

Presidente:

Maria de Fátima Coutinho Rodrigues, Prof. Doutor

Vogais:

António Cardoso Costa, Prof. Doutor

Ido Aharon Iurgel, Prof. Doutor

João Paulo Jorge Pereira, Prof. Doutor



Porto, Novembro 2010

“Animation can be incredibly fun, but it's also a very difficult art to master.”

in Digital Character Animation 2 (Volume 1)

Dedicado às minhas sobrinhas

Maria Leonor e Maria Beatriz.

I. AGRADECIMENTOS

A realização deste projecto não foi resultado de um trabalho individual: sem o apoio e a participação de uma série de pessoas nunca teria sido possível concretiza-lo. A todas elas queria dar os meus mais sinceros agradecimentos, nomeadamente:

- Ao Prof. Doutor João Paulo Jorge Pereira, na qualidade de orientador da minha dissertação de mestrado, pelo enorme apoio que me tem dado através dos seus conselhos, e na resolução das minhas dúvidas;
- Ao Instituto Superior de Engenharia do Porto (ISEP), especialmente ao Departamento de Engenharia Informática (DEI), e a todos os meus professores e colegas de curso, pela inesquecível experiência que tem sido estudar com eles;
- A todos os colaboradores do Centro de Computação Gráfica, especialmente ao Prof. Doutor Ido Aharon Iurgel, na qualidade de gestor do projecto *VirtualActor*, e a toda a equipa de desenvolvimento do projecto *VirtualActor*, pelo seu enorme apoio e encorajamento, e pelos momentos inesquecíveis que temos partilhado na realização do projecto;
- À minha família, especialmente aos meus pais, Luís e Rosa e aos meus irmãos Pedro e Ricardo, pelo afecto e apoio que sempre me têm oferecido;
- À minha cunhada, Regina, pelo apoio, tempo cedido e interesse demonstrado neste projecto;
- À minha namorada, Sílvia Jacob, pela paciência, pela atenção e pelo apoio dado ao longo deste projecto;
- Aos meus amigos, pela compreensão dos meus silêncios e ausências, pelas mensagens carinhosas, pela amizade incondicional;

II. RESUMO

O principal objectivo da animação de personagens virtuais é o de contar uma história através da utilização de personagens virtuais emocionalmente expressivos. Os personagens têm personalidades distintas, e transmitem as suas emoções e processos de pensamento através dos seus comportamentos (comunicação não verbal). As suas acções muitas das vezes constituem a geração de movimentos corporais complexos. Existem diversas questões a considerar quando se anima uma entidade complexa, tais como, a posição das zonas móveis e as suas velocidades. Os personagens virtuais são um exemplo de entidades complexas e estão entre os elementos mais utilizados em animação computacional.

O foco desta dissertação consistiu na criação de uma proposta de sistema de animação de personagens virtuais, cujos movimentos e expressões faciais são capazes de transmitir emoções e estados de espírito. Os movimentos primários, ou seja os movimentos que definem o comportamento dos personagens, são provenientes da captura de movimentos humanos (*Motion Capture*). As animações secundárias, tais como as expressões faciais, são criadas em *Autodesk Maya* recorrendo à técnica *BlendShapes*. Os dados obtidos pela captura de movimentos, são organizados numa biblioteca de comportamentos através de um grafo de movimentos, conhecido por *Move Tree*. Esta estrutura permite o controlo em tempo real dos personagens através da gestão do estado dos personagens. O sistema possibilita também a transição eficaz entre movimentos semelhantes e entre diferentes velocidades de locomoção, minimizando o efeito de arrastamento de pés conhecido como *footskate*. Torna-se assim possível definir um trajecto que o personagem poderá seguir com movimentos suaves. Estão também disponíveis os resultados obtidos nas sessões de avaliação realizadas, que visaram a determinação da qualidade das transições entre animações. Propõem-se ainda o melhoramento do sistema através da implementação da construção automática do grafo de movimentos.

Palavras-Chave

animação computacional, personagens virtuais, comunicação não verbal, captura de movimentos, transição entre animações, grafo de movimentos.

III. ABSTRACT

The main objective of the virtual characters animation is to tell a story through the use of emotionally expressive virtual characters. The characters have distinct personalities, and transmit their emotions and thought processes through their behaviors (nonverbal communication). Their actions often constitute the generation of complex body movements.

There are several issues to consider when animating a complex entity such as the position of movable parts and their velocities. The virtual characters are an example of complex entities and are among the most used in computer animation.

The focus of this thesis was the creation of a proposal of animation system of virtual characters, whose movements and facial expressions are able to transmit emotions and moods. The primary movements or in other words the movements that define the behavior of the characters come from motion capture. The secondary animations such as facial expressions are created in Autodesk Maya using the blendShapes technique. The data obtained by motion capture are organized in a library of behaviors through a graph of motion known as Move Tree. This structure allows for real-time control of the characters through the management of the characters state. The system also allows a smooth transition between similar movements and different speeds of locomotion minimizing the effect of dragging of feet known as footskate. It thus becomes possible to define a path that the character can follow with a smooth motion. Are also available the results of the performed evaluation sessions that aimed to determine the quality of the transitions between animations. It's further proposed improve the system through the implementation of the automatic construction of the motion graph.

Key-Words

computer animation, virtual characters, non-verbal communication, motion capture, transition between animations, move tree.

IV. ÍNDICE

I. AGRADECIMENTOS.....	v
II. RESUMO.....	vii
III. ABSTRACT.....	ix
IV. ÍNDICE.....	xi
V. ÍNDICE DE FIGURAS.....	xv
VI. ÍNDICE DE EQUAÇÕES.....	xvii
VII. ÍNDICE DE TABELAS.....	xix
VIII.ACRÓNIMOS.....	xxi
1 INTRODUÇÃO.....	1
1.1 CONTEXTUALIZAÇÃO.....	2
1.1.1 Breve apresentação do projecto “ <i>VirtualActor</i> ”.....	2
1.2 OBJECTIVOS.....	4
1.3 ORGANIZAÇÃO DO DOCUMENTO.....	4
2 CONCEITOS PRELIMINARES.....	5
2.1 ANIMAÇÃO TRADICIONAL.....	5
2.1.1 Princípios fundamentais da animação.....	7
2.2 ANIMAÇÃO COMPUTACIONAL.....	15
2.2.1 Visão geral sobre o processo de animação computacional.....	16
2.3 MODELAÇÃO.....	17
2.3.1 Modelação Hierárquica.....	18
2.4 TÉCNICAS BÁSICAS DE ANIMAÇÃO COMPUTACIONAL.....	19
2.4.1 <i>Keyframing and In-Betweening</i>	19
2.4.2 Interpolação Linear.....	20
2.4.3 Interpolação Curva.....	21
2.4.4 Desenvolvimento do conceito de Actor Virtual.....	22
2.5 ANIMAÇÃO DE PERSONAGENS VIRTUAIS.....	25
2.6 CONTROLO DE MOVIMENTO DE PERSONAGENS ANIMADOS.....	25
2.6.1 Cinemática directa (<i>forward kinematics</i>).....	26
2.6.2 Cinemática inversa (<i>inverse kinematics</i>).....	26
2.6.3 Captura de movimentos (<i>motion capture</i>).....	27
2.6.4 Técnicas de dinâmica do movimento.....	28
2.7 MoCap (captura de movimentos).....	29
2.7.1 Tipos de Sistemas MoCap.....	32

2.8	MANIPULAÇÃO DE INFORMAÇÃO MOCAP	37
2.8.1	Métodos Baseados em Física.....	37
2.8.2	Métodos Baseados em Interpolação	38
2.9	COMPOSIÇÃO E TRANSIÇÃO DE ANIMAÇÕES.....	38
2.10	INTERPOLAÇÃO DE ROTAÇÕES 3D	40
2.10.1	Ângulos de Euler	40
2.10.2	Matrizes de Rotação	42
2.10.3	Quaterniões.....	43
2.11	MÁQUINA DE ESTADOS FINITA	46
2.11.1	Motion Graphs.....	46
2.11.2	Move Trees.....	48
2.12	SUMÁRIO.....	49
3	TECNOLOGIAS	51
3.1	VICON iQ 2.5	52
3.2	AUTODESK MOTIONBUILDER	54
3.3	AUTODESK MAYA 2009	56
3.3.1	<i>Rigging</i> de personagens virtuais.....	57
3.3.2	Animação de expressões faciais	58
3.3.3	Exportação, de modelos e animações para <i>Ogre 3D</i>	59
3.4	<i>Ogre 3D</i> – MOTOR GRÁFICO 3D	61
3.4.1	Skeleton	62
3.4.2	Animation State	62
3.4.3	Tipos de animação suportadas pelo <i>Ogre 3D</i>	63
3.5	CEGUI.....	65
3.6	SUMÁRIO.....	66
4	SOLUÇÃO PROPOSTA.....	67
4.1	CAPTURA ÓPTICA DE MOVIMENTOS.....	67
4.2	COMPOSIÇÃO DE ANIMAÇÕES DE POSE.....	71
4.3	MISTURA DE DIFERENTES TIPOS DE ANIMAÇÕES.....	72
4.3.1	Composição de animações de pose e de esqueleto.....	72
4.3.2	Composição de múltiplas animações de esqueleto.....	73
4.4	MÓDULO DE LOCOMOÇÃO.....	76
4.4.1	Interpolação de Rotações 3D.....	78
4.5	MÁQUINA DE ESTADOS.....	79
4.6	SUMÁRIO.....	80
5	ANÁLISE DE RESULTADOS.....	81

5.1	SUMÁRIO	87
6	CONCLUSÕES.....	89
6.1	MODIFICAÇÕES SUGERIDAS/ TRABALHO FUTURO.....	90
	Referências Documentais	93
	ANEXO A.....	97
•	ETAPA 0 - Iniciar Sistema.....	97
•	ETAPA 1 - Preparação do cenário	97
•	ETAPA 2 - Calibração	99
•	ETAPA 3 - Captura.....	101
•	ETAPA 4 - Tratamento da informação capturada.....	105

V. ÍNDICE DE FIGURAS

Figura 1 - Interface da aplicação de ensaio experimental CREATOR.....	3
Figura 2 - Zoetrope.....	6
Figura 3 - Animação <i>Stop-Motion</i> Wallace and Gromit [WallaceAndGromit]	7
Figura 4 – Temporização [Whitaker and Halas, 2002]	8
Figura 5 - Suavizar do início e do fim.....	9
Figura 6 - arcos.....	9
Figura 7 – Antecipação [Whitaker and Halas, 2002]	10
Figura 8 - Exagero [Whitaker and Halas, 2002]	10
Figura 9 - Amassar e esticar [Lasseter, 1987].....	11
Figura 10 - Acção Secundária	11
Figura 11 - continuidade e acção sobreposta [Maestri, 1999].....	12
Figura 12 - Acção contínua	13
Figura 13 - Pose a pose [Whitaker and Halas, 2009]	13
Figura 14 - Enquadramento.....	14
Figura 15 - desenho sólido	14
Figura 16 - Expressividade do personagem (<i>Woody</i> no filme de animação <i>Toy Story 1</i>).....	15
Figura 17 - The Ultimate Human	17
Figura 18 - Exemplo de articulação de revolução (a) e articulação esférica (b)	18
Figura 19 - Representação simples do esqueleto hierárquico de uma perna.....	19
Figura 20 - Frames chave e frames intermédias.....	20
Figura 21 - Interpolação linear, representada sobre a forma de gráfico.....	21
Figura 22 - Interpolação curva	22
Figura 23 - Cinemática directa [Maestri, 1999]	26
Figura 24 - Cinemática inversa [Maestri, 1999].....	27
Figura 25 - Captura de performance do actor Steve Buscemi com MoCap para o filme <i>Monster House</i> [Whitaker and Halas, 2009]	27
Figura 26 - Técnicas de dinâmica do movimento	28
Figura 27 - Exemplo de utilização da técnica <i>rotoscoping</i> [Fleisher, 1917].....	30
Figura 28 - Animação computacional produzida através de Mocap	32
Figura 29 – Sistema MoCap Óptico.....	33
Figura 30 - Sistema MoCap Magnético [O'Brien, 2000].....	34
Figura 31 - Sistema MoCap Mecânico.....	35
Figura 32 - Transição por composição com interpolação linear	39
Figura 33 - Sistema de rotação <i>Pitch-Yaw-Roll</i>	41
Figura 34 - Esquema de um giroscópio com três eixos de liberdade	41
Figura 35 - Situação normal (a), <i>Gimbal Lock</i> (b)	42
Figura 36 - Interpolação recorrendo às técnicas <i>lerp</i> e <i>slerp</i>	44
Figura 37 - Comparação entre <i>lerp</i> (A) e <i>slerp</i> (B).....	45
Figura 38 - Máquina de estados finita [Lau and Kuffner, 2005].....	46
Figura 39- Motion Graph básico	47
Figura 40 - Processo de animação computacional	51
Figura 41 - Análise e edição de informação de movimento utilizando Vicon iQ	52
Figura 42 - Interface <i>Vicon iQ 2.5</i>	53
Figura 43 - manuseamento de informação MoCap	55
Figura 44 - Criação de uma "história" em <i>MotionBuilder</i>	56

Figura 45 - Rigging de personagens virtuais em <i>Maya 2011</i>	57
Figura 46 - Animação de expressões faciais	58
Figura 47 – Configuração de blend shapes.....	59
Figura 48 – Exportação de modelos	60
Figura 49 - Sistema Vicon MX.....	68
Figura 50 – (A) varinha de 240 mm; (B) varinha de calibração de 120 mm.....	69
Figura 51 - Distribuição de Marcadores	70
Figura 52 – Expressões faciais: (a) pose <i>neutra</i> ; (b) pose <i>feliz</i> ; (c) pose <i>surpreendido</i> ; (d) mistura pose <i>feliz</i> e <i>surpreendido</i>	71
Figura 53 - Construção de fonemas da língua inglesa, para a animação de diálogos.....	72
Figura 54 - Composição de <i>Skeletal Animation</i> com <i>Pose Animation</i>	73
Figura 55 - Composição de animações.....	74
Figura 56 - Transição entre duas animações através da interpolação da orientação de cada articulação	75
Figura 57 - Mistura de animações de esqueleto	76
Figura 58 - Ajuste da velocidade de <i>corrida</i>	77
Figura 59 - Ajuste das velocidades de <i>caminhada</i> e <i>corrida</i>	78
Figura 60 - Interpolação de rotações 3D utilizando SLERP.....	78
Figura 61 - Diagrama de estados	80
Figura 62 - Teste do sistema proposto integrado na aplicação <i>VirtualActor</i>	82
Figura 63 - Resultados das transições entre animações.....	84
Figura 64 – Transições com origem nas animações <i>walk</i> , <i>run</i> e <i>idle</i>	85
Figura 65 - Transição entre o estado <i>corrida</i> e <i>descanso</i>	86
Figura 66 - Transições com destino nas animações <i>walk</i> , <i>run</i> , <i>idle</i>	86
Figura Anexo - 1 - Vicon Mx Box	97
Figura Anexo - 2 - Interface Vicon iQ 2.5	97
Figura Anexo - 3 - Arranque do sistema	97
Figura Anexo - 4 - Marcador.....	97
Figura Anexo - 5 - Disposição do cenário.....	98
Figura Anexo - 6 - Câmara Vicon MX.....	98
Figura Anexo - 7 - Definição da região de captura	99
Figura Anexo - 8 – Arranque da calibração das câmaras	99
Figura Anexo - 9 - Calibração das câmaras.....	100
Figura Anexo - 11 - Esquadro com marcadores	100
Figura Anexo - 10 - Calibração do chão.....	101
Figura Anexo - 12 - Resultado final após a calibração.....	101
Figura Anexo - 13 - Pormenor do pé e mão	101
Figura Anexo - 14 - Disposição dos marcadores, vista frontal e vista de retaguarda.....	102
Figura Anexo - 15 - Distribuição de Marcadores	102
Figura Anexo - 16 - Localização dos marcadores	103
Figura Anexo - 17 - Reconstrução da informação.....	104
Figura Anexo - 18 - Calibração do sujeito	104

VI. ÍNDICE DE EQUAÇÕES

(Equação 1)	20
(Equação 2)	40
(Equação 3)	42
(Equação 4)	43
(Equação 5)	43
(Equação 6)	43
(Equação 7)	44

VII. ÍNDICE DE TABELAS

Tabela 1 - Classificação qualitativa de transições entre animações	81
Tabela 2 - Resultados obtidos nas simulações (transições entre animações)	83

VIII. ACRÓNIMOS

3D	- Tridimensional
API	- Application Programming Interface
CCG	- Centro de Computação Gráfica
CEGUI	- Crazy Eddie's GUI System
FCT	- Fundação para a Ciência e a Tecnologia
FPS	- Frames por Segundo
FSM	- Finite-State Machine (maquina de estados finita)
GUI	- Graphical User Interface (Interface Gráfico do Utilizador)
IA	- Inteligência Artificial
I & D	- Investigação e Desenvolvimento
IK	- Inverse Kinematics (cinemática inversa)
ISEP	- Instituto Superior de Engenharia do Porto
LED	- Light Emitting Diode (diodo emissor de luz)
LERP	- Interpolação linear
MoCap	- Motion Capture
OGRE	- Object Oriented Graphics Rendering Engine
OpenGL	- Open Graphics Library
SLERP	- Interpolação linear esférica

1 INTRODUÇÃO

A animação computacional constitui um elemento vital de variadas aplicações tais como os jogos, os filmes e os simuladores de ambientes virtuais. Permite ao utilizador obter sensações através da simulação de diversos ambientes melhorando a qualidade da experiência. Num jogo é muito importante animar adequadamente os movimentos corporais humanos dos seus personagens, de modo a tornar o jogo mais atractivo, e de forma a proporcionar uma experiência realista.

Para além de serem complexos, os movimentos humanos têm um papel importante na ciência cognitiva. No quotidiano, movimentos tais como gestos são a forma de comunicação mais utilizada logo a seguir à fala. Os movimentos podem ser utilizados como uma forma directa de comunicação. Os humanos extraem significados e sensações, que são inconscientemente incluídos nos movimentos de outros indivíduos [Hugill et al., 2010]. Portanto, se a animação de um personagem dá ao utilizador a sensação de alegria, e era suposto o personagem estar triste, a coerência da animação é afectada de forma negativa. Assim, a animação de personagens humanos requer uma atenção muito especial.

Criar animações de personagens com comportamento humano é uma tarefa difícil. Se a animação a ser criada contiver padrões complexos tal como a *caminhada*, o trabalho a ser feito pelo animador torna-se ainda mais exigente. Movimentos tais como caminhar têm tendência para ser repetitivos. Assim, estes podem ser criados de forma potencialmente mais eficiente. Neste contexto, rever os métodos actuais para a criação de animações de personagens humanos e identificar as diferentes secções do processo de animação ganha bastante relevância. Um personagem virtual com características humanas deve

locomover-se de forma realista. Uma forma de abordar a questão do controlo da locomoção é a utilização de cliques de capturas de movimentos, cuja informação esteja estruturada num grafo de movimentos.

1.1 CONTEXTUALIZAÇÃO

A investigação e desenvolvimento obtido no âmbito desta dissertação está integrado no projecto *VirtualActor*, que se encontra em fase de desenvolvimento no Centro de Computação Gráfica (CCG) situado no Campus de Azurém da Universidade do Minho, em Guimarães. Os esforços foram centrados na criação de um sistema de animação de personagens virtuais, aproveitando funcionalidades já existentes e adaptando-as a personagens de corpo inteiro. Os personagens deverão estar dotados com a capacidade de representar comportamentos e emoções específicos de um actor humano.

1.1.1 Breve apresentação do projecto “*VirtualActor*”

O projecto *VirtualActor* desenvolve a base de software para a criação de actores virtuais. Este projecto, de I & D (Investigação e Desenvolvimento), foi financiado pela FCT (Fundação para a Ciência e a Tecnologia) com a referência PTDC/EIA/69236/2006, e envolve programadores e investigadores do CCG e da Universidade do Minho.

A meta é a criação de actores virtuais, que podem ser utilizados de forma semelhante a um actor real. O utilizador transforma-se no realizador do filme de animação. Ele dirige os actores virtuais com comandos simples tais como “*caminha na direcção do personagem x manifestando alegria*”. Pretende-se assim que o realizador crie de forma simples os seus filmes de animação, e que estes correspondam às suas intenções.

O projecto visa desenvolver um quadro de software genérico, que inclui editores e um sistema de animação para personagens virtuais. O quadro dos editores vai permitir o uso de diferentes técnicas de controlo e de IA (Inteligência Artificial) para actores virtuais sempre no mesmo quadro conceptual e de edição. Aplicações desse tipo requerem humanos virtuais autónomos e inteligentes, com capacidade de adaptação, percepção e memória. Com a ajuda dos resultados do projecto, cada utilizador poderá criar filmes de animação em 3D com actores virtuais emocionalmente expressivos, ou seja, poderão

transmitir ilusão de vida, aptidão para agir com autonomia e para se expressarem emocionalmente, com voz e com sinais não verbais. Esta tecnologia apresenta-se vantajosa, visto não ser necessário despender longos períodos de formação nem de muito investimento. Qualquer utilizador será capaz de ser o realizador do seu próprio filme.

Para além das vantagens que os resultados trazem para a produção de filmes de animação, os actores virtuais podem ser utilizados em variadíssimas outras áreas, tais como jogos, na educação, em histórias interactivas, e também na terapia de fobias sociais e no treino comportamental, como por exemplo, para conduzir certos tipos de diálogos [VirtualActor, 2008].

Aquando do início desta dissertação, o projecto *VirtualActor* já estava a decorrer, tendo já uma aplicação de ensaio experimental em *Ogre 3D* (CREATOR) onde os actores virtuais eram simples cabeças falantes. Na Figura 1 é possível visualizar a interface da aplicação de ensaio experimental CREATOR.

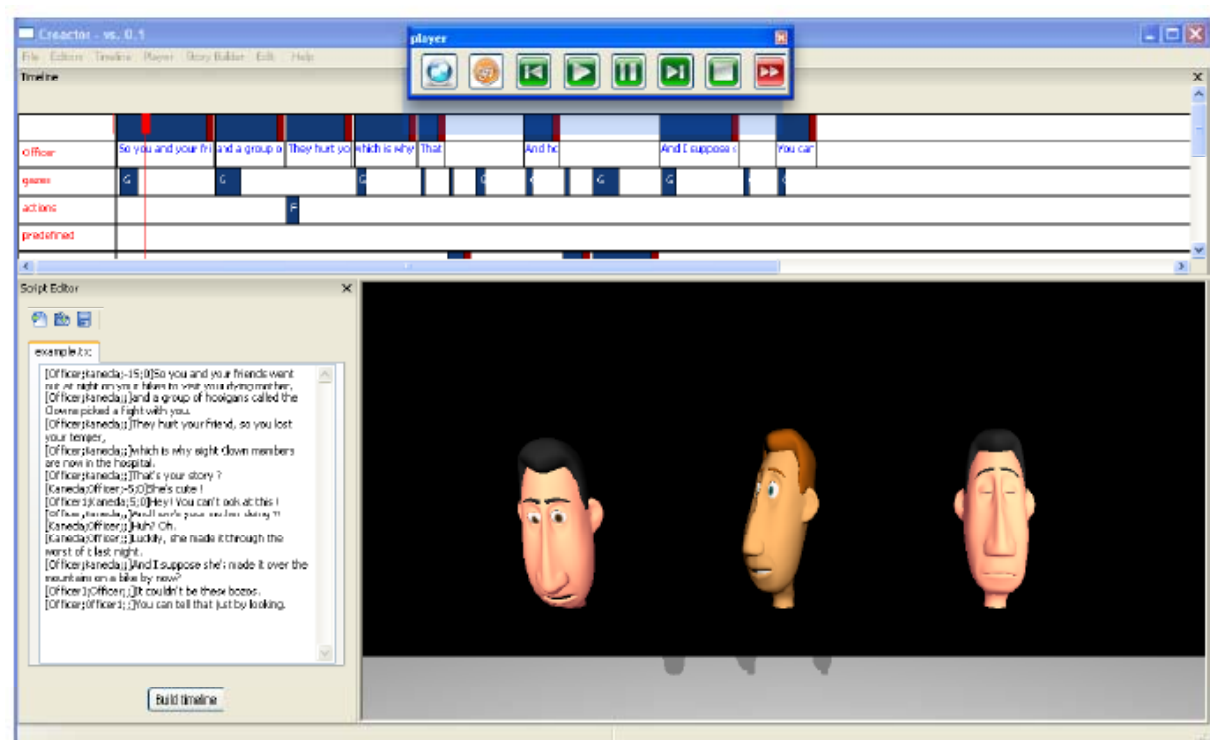


Figura 1 - Interface da aplicação de ensaio experimental CREATOR

1.2 OBJECTIVOS

O tema focado nesta dissertação consiste no desenvolvimento de um sistema de animação de personagens virtuais para comunicação não verbal, em *Ogre 3D*.

O objectivo da presente dissertação será o desenvolvimento de um protótipo de um sistema para a simulação de movimentos humanos realistas em tempo real, através da combinação de *Motion Capture* com sistemas dinâmicos. Através da construção de animações de personagens humanos de forma simples e sem a necessidade de despendere um tempo de aprendizagem significativo seria possível a criação de muitas aplicações novas e apelativas. Seria interessante que um utilizador principiante tivesse a capacidade de sintetizar o movimento humano introduzindo somente algumas instruções simples.

Abordar este problema, da animação de actores virtuais, implica a representação da expressão de emoções ou comportamentos de objectos vivos através da utilização de movimento. Aproveitando as funcionalidades oferecidas pelo motor gráfico *Ogre 3D*, pretende-se um sistema de animação de actores virtuais capaz de sintetizar em tempo real alguns movimentos, de forma a transmitir as expressões faciais e corporais características de um actor real. Para isto deverá permitir a combinação de sequências de animações, de forma a criar transições suaves entre animações. Pretende-se que este sistema seja a base para o desenvolvimento de actores virtuais.

1.3 ORGANIZAÇÃO DO DOCUMENTO

A estrutura da dissertação encontra-se dividida em seis capítulos. O presente capítulo introduz e contextualiza o trabalho em questão no projecto *VirtualActor* e apresenta os objectivos. O segundo capítulo destina-se a introduzir, de forma sucinta, diversos conceitos fundamentais associados à animação de personagens. No capítulo seguinte, é dada uma visão geral sobre algumas das tecnologias e ferramentas utilizadas. O capítulo quatro apresenta a solução proposta, dando especial destaque à captura de movimentos e à composição de animações. No capítulo cinco são analisados os resultados experimentais obtidos nas diferentes simulações. O capítulo final, apresenta as conclusões finais e possíveis direcções a tomar em desenvolvimentos futuros.

2 CONCEITOS PRELIMINARES

Neste capítulo são introduzidos alguns conceitos necessários à compreensão dos capítulos seguintes e ao âmbito geral da dissertação. Estes incluem as técnicas de animação tradicional e a forma como estas influenciaram as técnicas de animação computacional. Será também introduzida a temática da captura de informação de movimento e os processos de tratamento dessa informação.

Os primeiros livros e filmes animados foram criados na volta do séc. XIX, mas a maioria dos princípios de animação foram desenvolvidos durante as duas primeiras décadas do séc. XX e aperfeiçoadas com as animações *cartoon* desenhadas à mão dos anos 30 e 40. Algumas das técnicas de animação computacional usadas para criar sequências de imagens estáticas são baseadas em técnicas tradicionais de animação, outras são simplesmente simulações baseadas em computador de mundos tridimensionais e personagens em movimento [Kerlow, 2004].

2.1 ANIMAÇÃO TRADICIONAL

Segundo [Lever, 2002] a animação é mágica. Visualizar os resultados de um personagem a ganhar vida, caminhando, conversando, ou lutando, gera emoções que são a verdadeira recompensa para o trabalhoso processo de animação. Animar significa dar vida a um objecto, imagem ou desenho inerte.

A animação é a arte do movimento expresso com imagens que foram retiradas directamente da realidade.

Uma frame consiste numa única imagem estática. As animações são compostas por milhares de frames, mas a unidade de animação é a frame. A apresentação rápida de sequências de várias imagens estáticas (frames), permite transmitir a ilusão de movimento. As frames que compõem uma animação são substituídas tão rapidamente que o espectador é incapaz de detectar a mudança das imagens, lendo o resultado como uma sequência de movimento. A Animação aproveita uma limitação dos nossos olhos, a persistência da visão. Os nossos olhos são óptimos na detecção de movimento, mas existe um limite finito para a rapidez com que podemos ver uma cena a mudar. No século XIX foram inventados vários aparelhos que exploram esta limitação. A Figura 2 ilustra um desses aparelhos, conhecido como *Zoetrope*, que permite ao utilizador visualizar uma sucessão de imagens no interior de um cilindro, através de buracos realizados na sua superfície. Foi descoberto que, se mostrarmos uma série de imagens estáticas com rapidez suficiente, então o resultado aparece aos nossos olhos como uma cena em movimento. O movimento de um *Zoetrope* cria filmes de animação, tendo sido uma importante forma de entretenimento e um negócio lucrativo para os seus criadores [Lever,2002].

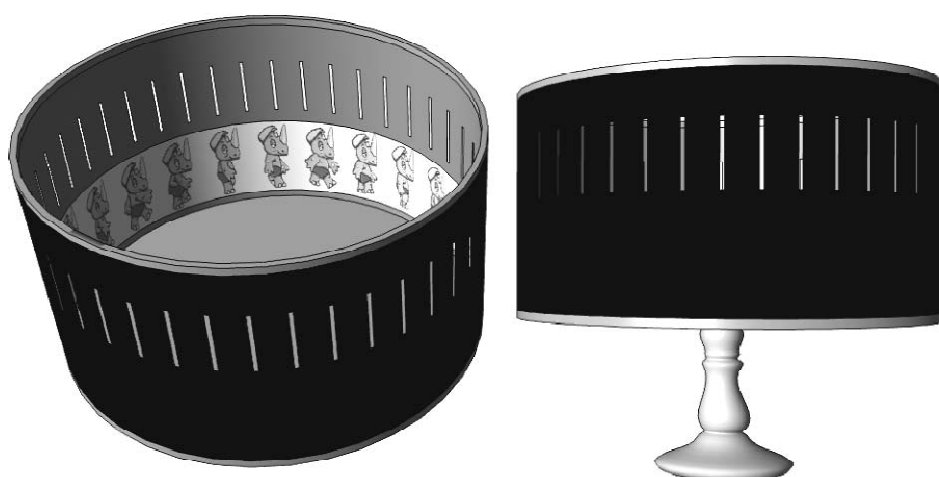


Figura 2 - Zoetrope

Com a criação das principais companhias de animação, como a *Walt Disney*, foram investidos esforços no sentido de tornar o processo de animação melhor. No início, os animadores desenhavam as cenas completas no papel, que depois fotografavam para formar um filme. Mais tarde, os animadores começaram a usar folhas transparentes, denominadas *cells*, e desenhavam os diferentes elementos da cena em folhas diferentes. Isto permitiu a criação de uma aproximação em camadas em que elementos estáticos, como o fundo, eram desenhados uma vez e elementos dinâmicos eram animados recorrendo a diferentes folhas. Esta técnica, também chamada “*cel animation*”, dominou a animação em duas

dimensões, até surgirem computadores com capacidade de processamento suficiente para desempenhar as mesmas funções.

Na Figura 3 está ilustrado um método que permite criação de animações através do posicionamento de modelos reais, que são posteriormente fotografados. Esta técnica ficaria conhecida como animação “*stop-motion*”. Apesar de este método requerer paciência e eficácia foi utilizado em variados filmes de animação e continua a ser utilizado actualmente.



Figura 3 - Animação *Stop-Motion* Wallace and Gromit [WallaceAndGromit]

2.1.1 Princípios fundamentais da animação

O realizador da *Pixar*, Jonh Lasseter escreveu um artigo [Lasseter, 1987] em que descreve a aplicação dos princípios da animação tradicional, originalmente desenvolvidos pelos animadores dos estúdios da *Walt Disney* na década de 30, à animação de personagens virtuais.

Segundo este autor quer a animação seja gerada manualmente ou através de computador, o primeiro objectivo do animador é cativar a atenção do espectador. O animador deve ter um conceito claro do que entretém a audiência, e as ferramentas e capacidades para tornar as ideias claras. Para Jonh Lasseter as ferramentas não são apenas o hardware e software. Existem princípios fundamentais da animação que também devem ser utilizados como ferramentas para criar boas animações. Os princípios fundamentais da animação tornaram-se a base do desenho manual de personagens animados, continuando a ser relevantes actualmente, pois ajudam no processo de criação de personagens e de situações mais credíveis. Os doze princípios fundamentais da animação são os seguintes: temporização; suavizar do início e do fim; arcos; antecipação; exagero; amassar e esticar;

acção secundária; continuidade e acção sobreposta; acção contínua e pose a pose; enquadramento; desenho sólido; apelo.

2.1.1.1 Timing (temporização):

O tempo (ou velocidade) de uma acção é a essência de uma animação. A velocidade a que algo se movimenta traduz o que o objecto é e o porquê de se estar a movimentar. Algo como um piscar de olhos pode ser rápido ou lento. Se for rápido, o personagem parecerá alerta; se for lento, poderá parecer cansado. Os personagens utilizados na criação de *cartoons* são normalmente caracterizados pela sequenciamento rápido entre poses. As animações reais tendem a ter uma temporização mais lenta, conferindo-lhe mais veracidade. Portanto, manter a concatenação dos eventos e sua coerência temporal são fundamentais para a autenticidade da animação. A Figura 4 ilustra um gráfico da velocidade de um objecto a cair por acção da força gravítica. Verifica-se que, neste caso, a distância percorrida vai aumentando ao longo do tempo; ou seja, existe aceleração.

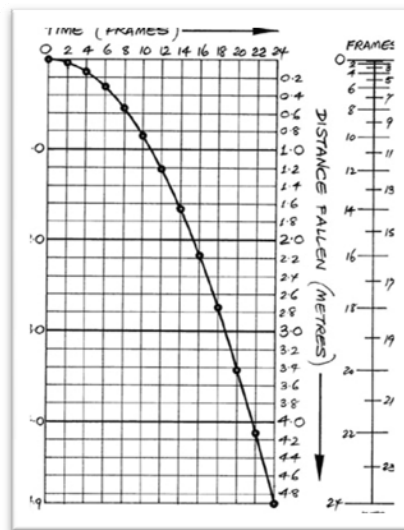


Figura 4 – Temporização [Whitaker and Halas, 2002]

2.1.1.2 Slow-in and Slow-out (suavizar do início e do fim)

Um princípio básico de dinâmica aplicada à animação é que os objectos precisam de tempo para iniciar e para finalizar o movimento. São poucos os objectos ou elementos que já começam um movimento na velocidade final. Na maioria dos casos é necessário um tempo de aceleração e desaceleração para atingir o movimento ou o repouso. Logo este princípio consiste na desaceleração do início e do final da acção. Esta técnica realça a zona central da acção, permitindo que o espectador

se adapte às mudanças da cena. Na Figura 5 é possível visualizar um exemplo da aplicação deste princípio. Consiste numa bola a saltar, a qual tende a ter uma velocidade maior quando está a cair do que quando está a subir. Isto deve-se ao facto de a bola estar sujeita à aceleração da gravidade.

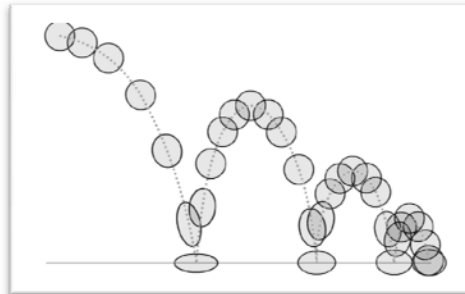


Figura 5 - Suavizar do início e do fim

2.1.1.3 Arcs (arcos)

A utilização de arcos para animar o movimento de personagens permite obter um aspecto natural, pois no mundo real quase todas as acções são feitas segundo linhas precisas e harmoniosas que permitem uma suavização dos movimentos. Quando se criam animações devem ser construídos movimentos com trajectórias curvas em vez de lineares. Se, ao animar o movimento de um personagem, não for respeitado este princípio, o seu movimento será pouco natural. A Figura 6 pretende representar a trajectória do corpo do personagem a descrever um arco.

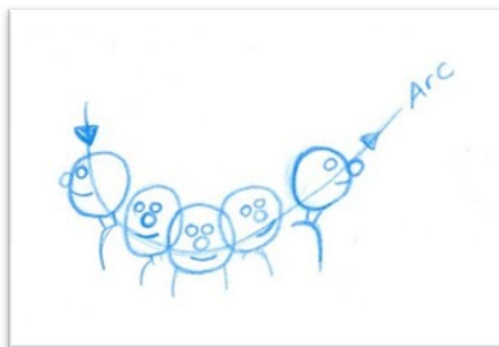


Figura 6 - arcos

2.1.1.4 Anticipation (antecipação)

As acções numa animação normalmente acontecem em três etapas: A organização do movimento, a acção propriamente dita e o seguimento da acção. A primeira etapa é conhecida como antecipação. Em alguns casos, a antecipação é essencial por razões de natureza física. Por exemplo, para arremessar um objecto pesado é necessário balançar o braço para trás. Esta acção do braço é a antecipação, enquanto o lançamento é o movimento propriamente dito. A antecipação é usada para chamar a atenção do

espectador e para prepará-lo para a etapa seguinte. Para acções mais rápidas é geralmente necessária uma fase de antecipação mais longa. A Figura 7 está dividida em três fases (1, 2 e 3), e pretende representar a preparação para agarrar algo numa mesa. Na fase 1 o personagem observa o alvo em repouso; na segunda fase antecipa o movimento seguinte; na terceira fase agarra o objecto.

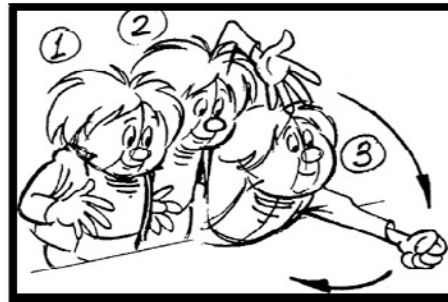


Figura 7 – Antecipação [Whitaker and Halas, 2002]

2.1.1.5 Exaggeration (exagero)

O exagero é utilizado para acentuar uma acção. Deve ser utilizado de maneira cuidadosa e equilibrada, e nunca arbitrariamente. O resultado deverá ser uma animação realística e divertida. Os exageros, as dimensões anormais e a desproporção são mais apropriados para animar personagens *cartoon*. Na Figura 8 está ilustrado o mesmo movimento de duas formas diferentes. A Figura 8 (B) representa uma tacada de golfe onde o movimento do personagem é exagerado, ao contrário do que acontece na Figura 8 (A).

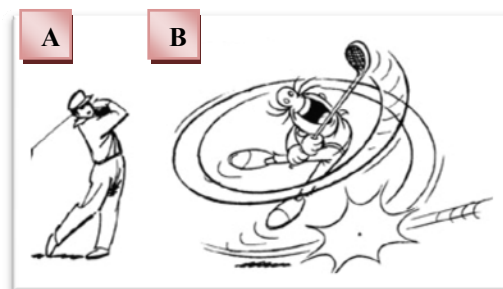


Figura 8 - Exagero [Whitaker and Halas, 2002]

2.1.1.6 Squash and Stretch (amassar e esticar)

Este é um dos princípios base da animação, que todos os elementos envolvidos na produção de animações devem dominar. Tem como função mostrar pela deformação de um objecto o seu peso e flexibilidade. A distorção de objectos ou de personagens numa cena acentua o movimento e o realismo da animação. Através deste princípio podem ser realçados atributos físicos como a inércia, o peso ou a

velocidade. Por exemplo, se uma bola de borracha salta e bate no chão, a mesma sofre uma força fazendo com que o seu corpo seja esmagado. A utilização mais directa deste princípio na animação de personagens é a contracção dos seus músculos. Como se pode visualizar na Figura 9 os objectos rígidos podem também, de certo modo, ser esmagados e esticados: basta pensar no caso de um objecto que possua articulações no seu corpo, como um candeeiro de mesa articulado. O seu corpo é rígido, mas pode ser animado (deslocando-se, por exemplo aos saltos). Neste caso, o seu corpo deverá ser esmagado e esticado, respeitando as suas articulações.

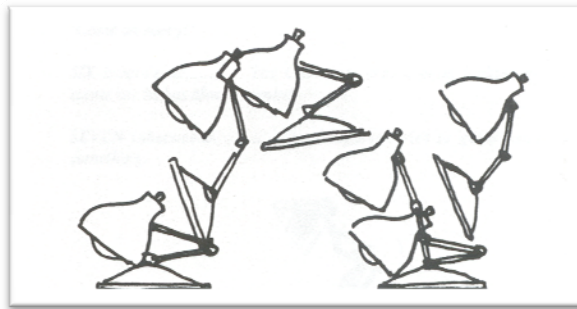


Figura 9 - Amassar e esticar [Lasseter, 1987]

2.1.1.7 *Secondary Action* (acção secundária)

Acção secundária é uma acção que é resultante de outra acção. As acções secundárias são importantes para aumentar o interesse e acrescentar complexidade à animação. Este tipo de acções enriquece a acção principal e acrescenta mais dimensão ao personagem, reforçando a acção principal. Por exemplo, durante o galopar de um cavalo a acção principal é feita com as patas, mas o oscilar da cauda (Figura 10), o abanar da cabeça e todas as outras acções do corpo são acções secundárias, ou acções de suporte.



Figura 10 - Acção Secundária

2.1.1.8 *Follow Through and Overlapping Action* (continuidade e acção sobreposta)

Este princípio é aplicado quando o personagem muda de direcção, enquanto as suas roupas ou cabelos continuam a movimentar-se de acordo com o movimento anterior. Algumas frames após a mudança de

direcção do personagem as suas roupas, cabelos e outros apêndices devem segui-lo, segundo a nova direcção. Pequenas variações na temporização e velocidade dos elementos que compõem o personagem tornam o movimento mais natural e interessante. Deve garantir-se que uma acção nunca para por completo antes de ser iniciada outra acção.

A Figura 11 é composta por quatro fases (A, B, C e D) e representa o princípio que consiste em manter a acção fluida, sem interrupções abruptas de movimentos. Na fase A o cão ainda não parou de caminhar; na fase B as patas da frente são as primeiras a parar; na fase C as suas orelhas e cauda sobrepõem o movimento da animação seguinte. Finalmente na fase D, o cão já se encontra na animação de *descanso*.

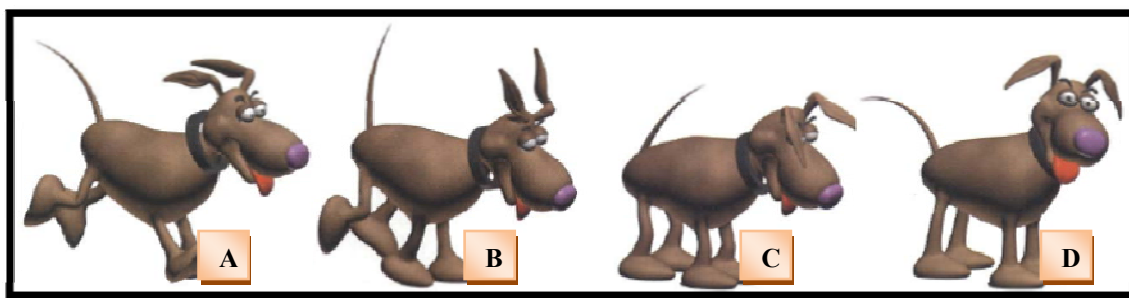


Figura 11 - continuidade e acção sobreposta [Maestri, 1999]

2.1.1.9 *Straight Ahead Action and Pose-to-Pose Action (acção contínua e acção pose a pose)*

A acção contínua e a acção pose a pose têm como base duas técnicas de animação que obtêm resultados bastante diferentes. No início da animação desenhada à mão, a animação pose a pose tornou-se a técnica de animação mais utilizada, devido à capacidade de quebrar o movimento de forma estruturada numa série de poses chave bem identificadas. Na acção contínua o personagem movimenta-se espontaneamente passo a passo até terminar a acção.

- Acção contínua: Neste método o animador começa por desenhar a primeira frame, e vai desenhando uma frame de cada vez até ao fim da cena, sem necessidade de um planeamento prévio. Com este método pode perder-se a noção das proporções, mas ganha-se espontaneidade. A Figura 12 ilustra o processo de animação recorrendo a este método, que é o ideal para cenas mais bruscas, com a vantagem de manter a criatividade do artista.



Figura 12 - Acção contínua

- Pose a pose: É uma técnica que permite um maior planeamento, pois é guiada por desenhos-chave. Este método torna mais simples o controlo das proporções. O animador tem em consideração quais os desenhos que serão necessários para animar a cena, desenhando as frames chave, as quais serão completadas, mais tarde, com o desenho das frames intermédias. A técnica pose a pose ilustrada na Figura 13 é usada em animações onde é importante a transição suave entre as poses. No entanto, por vezes, uma animação é criada recorrendo a ambos os métodos de animação.



Figura 13 - Pose a pose [Whitaker and Halas, 2009]

2.1.1.10 Staging (enquadramento)

Trata-se da apresentação de uma ideia de forma a torna-la inteligível para o espectador. Durante uma acção deve transparecer claramente a atitude, o estado de espírito e a reacção ou do personagem. A forma como a cena é estruturada é importante para dar continuidade à linha da narrativa. A Figura 14 ilustra a linguagem corporal de dois personagens sentados a uma mesa. Enquanto na Figura 14 (B), o facto de os personagens estarem próximos indica que estão concentrados um no outro, na Figura 14 (A) o facto de um personagem estar inclinado e o outro a olhar para longe mostra que existe algum desconforto entre ambos.

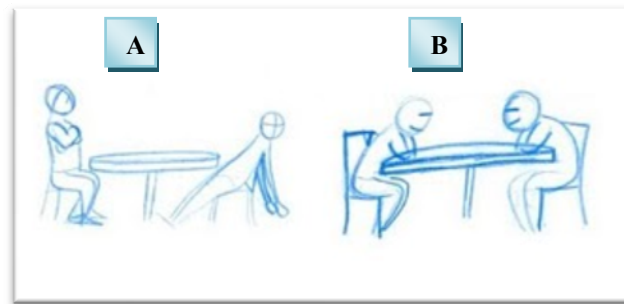


Figura 14 - Enquadramento

2.1.1.11 Solid drawing (desenho sólido)

O artista deve criar uma figura volumétrica, sólida e tridimensional, aplicando na imagem a percepção de peso, profundidade e equilíbrio. Devem ser evitadas poses compostas por simetrias, ou seja não se deve espelhar as posições do rosto, braços ou pernas, de forma a evitar uma apresentação pouco natural. Como se pode visualizar na Figura 15 o movimento corporal raramente é simétrico, embora a postura global deva ser equilibrada.



Figura 15 - desenho sólido

2.1.1.12 Appeal (apelo)

Enquanto um actor tem carisma, um desenho animado é emocionalmente expressivo. Mas isto não significa que o personagem precise apenas de ter um aspecto agradável. Todos os personagens têm de ter uma personalidade atraente sejam eles heróis, ou vilões; cómicos ou meigos. Este princípio pressupõe que o personagem inclui um *design* fácil de ser lido e um desenvolvimento da personalidade capaz de capturar e envolver o interesse do espectador. Um personagem tem de ser único pois, tal como os humanos, ninguém é igual a ninguém. Isto dá a cada personagem um estilo próprio, e a capacidade de reagir às situações de forma distinta. Na Figura 16, o personagem *Woody* do filme de

animação *Toy Story* é um exemplo de personagem animado capaz de cativar o público através das suas características reais e humanas.



Figura 16 - Expressividade do personagem (*Woody no filme de animação Toy Story 1*)

O movimento pode acontecer por vários motivos, mas todos os movimentos têm características em comum. Todos os objectos se movem porque são afectados por forças e todos os objectos em movimento estão sujeitos à inércia que faz com que seja difícil mudar a sua trajectória.

2.2 ANIMAÇÃO COMPUTACIONAL

Na década de 60, o computador começou a ser utilizado como ferramenta de apoio à criação de filmes pelo processo de animação tradicional. Este facto desencadeou o que hoje constitui uma das maiores linhas de investigação da computação gráfica, a animação por computador. O processo de animação começou a alterar-se gradualmente. Os computadores começaram por ser utilizados apenas para desempenhar tarefas anteriormente realizadas por animadores com menos experiência.

À medida que as capacidades dos computadores foram aumentando, os ambientes virtuais compostos por geometria 3D passaram a poder ser representados utilizando computadores. Projecções bidimensionais de cenas 3D foram criadas utilizando uma aproximação semelhante à da fotografia. Este avanço ajudou a criar a técnica de *stop-motion* virtual, chamada animação 3D. A criação e posicionamento de modelos virtuais tridimensionais tornou-se muito mais fácil e coerente que criar “fantoques” e posicioná-los, de forma a obter animações 3D *stop-motion*.

Hoje, a animação industrial é maioritariamente dominada por animações tridimensionais, mas ainda existem animações 2D populares a serem produzidas, tais como as Japonesas *animés*.

O número de frames que constituem um segundo de animação depende do sistema a que se destina a animação. Um segundo de animação à velocidade normal em vídeo são 30 frames; um segundo num filme contém 24 frames. Em computação interactiva em tempo real varia consoante o desempenho do hardware, variando entre 8 e 60 frames. O número de frames por segundo de uma animação é também chamado de taxa de exibição e normalmente indicado com a abreviatura fps [Kerlow, 2004].

2.2.1 Visão geral sobre o processo de animação computacional

O processo de criação de animação computacional é baseado em factores tais como objectivos criativos, orçamento, calendarização e recursos. Na prática, o processo de produção adopta diferentes formas e variações, mas as etapas básicas incluem a pré-produção, produção e pós-produção [Kerlow, 2004].

A pré-produção envolve todo o planeamento que tem lugar antes de ser produzida a animação computacional. Esta etapa inclui tarefas tais como o desenvolvimento do guião e o desenvolvimento da base do aspecto visual do projecto. A pré-produção é o alicerce do projecto. Uma pré-produção errónea ou inadequada resulta geralmente em atrasos e custos imprevistos.

De forma simplificada, a etapa de produção envolve uma série de passos comuns, tais como modelação, *rigging*, animação e *rendering*. Primeiro são modelados os personagens, os objectos e o ambiente a utilizar na animação computacional tridimensional. Na fase de modelação pode recorrer-se a ferramentas de modelação 3D para esculpir os objectos, ou a um digitalizador para capturar a forma do modelo físico. Uma vez modelados os personagens virtuais, estes podem ser combinados com um esqueleto, o que simplificará o processo de animação. A este processo dá-se o nome de *rigging*. Na fase de animação podem ser utilizadas diversas técnicas que serão introduzidas nas próximas secções. Depois de modelados e animados, os objectos são renderizados. O *rendering* é o processo de representação virtual de modelos animados, onde são calculadas as imagens computacionais finais.

Uma vez renderizadas as imagens, podem ser aplicadas a estas várias técnicas de pós-produção. Por exemplo, as imagens geradas podem ser compostas ou misturadas com outras imagens. A animação computacional pode ainda ser distorcida, retocada, ou corrigida ao nível das cores.

Apesar de esta descrição do processo de animação computacional estar dividida em etapas bem definidas, este processo nunca é inteiramente linear, pois várias tarefas são independentes, podendo ser sobrepostas ou acontecer paralelamente.

2.3 MODELAÇÃO

A criação de animações tridimensionais para entidades complexas não é um objectivo fácil de alcançar. As tarefas a realizar ao longo do processo de animação dependem fortemente da complexidade do personagem. Conforme o personagem se torna mais detalhado, o processo de modelação torna-se mais trabalhoso e culmina num modelo com elevado número de polígonos. A animação de tais entidades complexas é também uma tarefa difícil desde que o número de regiões que se possam mover aumente com a complexidade da entidade. O animador terá de considerar todas estas partes móveis e criar animações convincentes para estas. A forma como estas entidades complexas se movem é, também, normalmente complexa.

Independentemente do método de representação, os modelos humanos realistas são detalhados e consequentemente difíceis de criar. Na Figura 17 é possível visualizar um excerto de um dos melhores modelos humanos existentes, conhecido como *The Ultimate Human*. Foi criado pela cgCharacter [cgCharacter] e contém 772 500 polígonos. Contudo, para a maioria das aplicações, modelos com um nível de detalhe mais baixo serão suficientes, os modelos humanos requerem normalmente um elevado número de objectos geométricos a que geralmente se dá o nome de primitivas. O número de primitivas no modelo depende do nível de detalhe e o nível de detalhe determina o realismo do modelo.

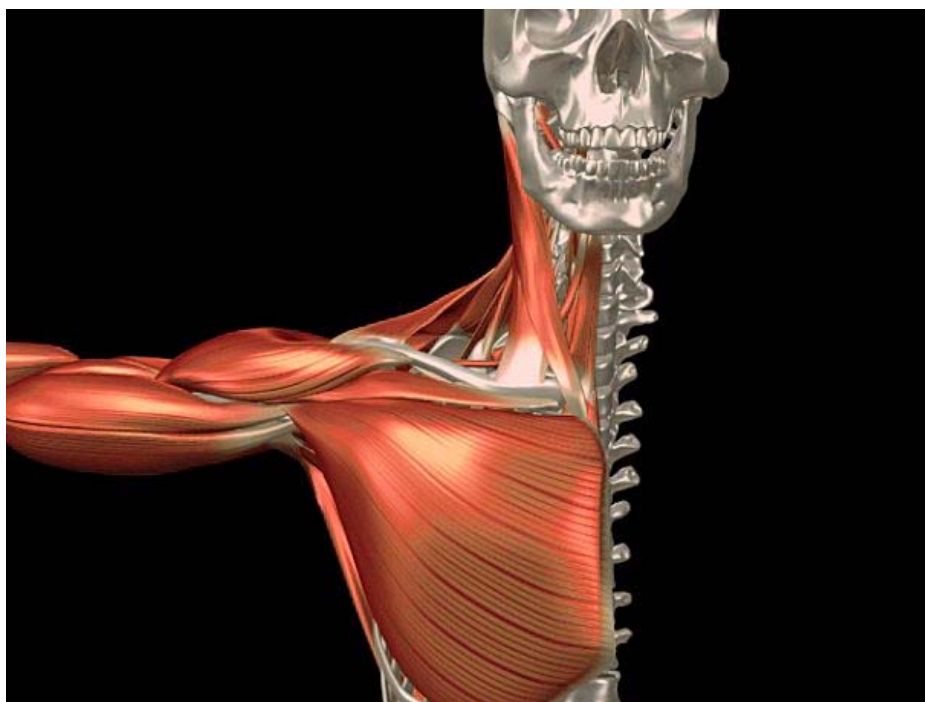


Figura 17 - The Ultimate Human

2.3.1 Modelação Hierárquica

Os modelos humanos incluem um elevado número de primitivas bem como um elevado número de regiões deformáveis (articulações). Para representar o movimento de criaturas vivas tais como humanos, animais ou insectos, é conveniente armazenar as relações entre cada parte móvel de um personagem. A manipulação destes modelos complexos requer aproximações sistemáticas em vez da manipulação de cada primitiva separadamente. Por exemplo, quando o animador pretende levantar o braço do modelo, ele deve ser capaz de o fazer facilmente sem se preocupar com cada primitiva que necessita de ser movida ou deformada. Para tal é necessário adicionar um esqueleto e controlos ao modelo de modo a permitir a manipulação do personagem. Este esqueleto é composto por articulações que impõem a hierarquia do modelo. As articulações do corpo humano têm estruturas extremamente complexas, sendo a sua modelação computacional complexa e morosa. De forma a simplificar e a aumentar a eficiência computacional, as articulações biológicas são, na grande maioria dos casos, substituídas pela utilização de articulações mecânicas de geometria ideal, tais como a articulação de revolução e a articulação esférica, representadas na Figura 18(A) e Figura 18 (B), respectivamente.

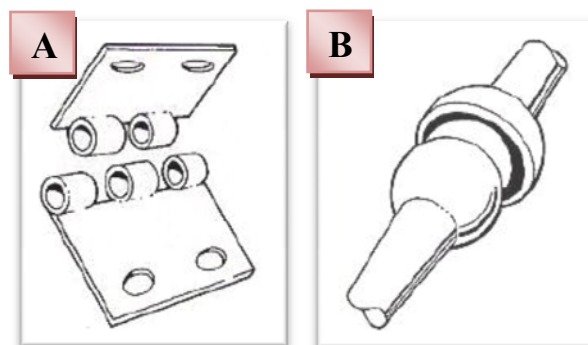


Figura 18 - Exemplo de articulação de revolução (a) e articulação esférica (b)

A Figura 19 ilustra um esqueleto hierárquico de uma perna onde os ossos, anca, canela e pé estão ligados pelas articulações joelho e tornozelo. O osso anca é o osso de nível superior na hierarquia e o seu movimento afecta automaticamente a canela e o pé. Desta forma simples, os ossos que estão em níveis inferiores na hierarquia são coordenados pelos ossos de nível superior.

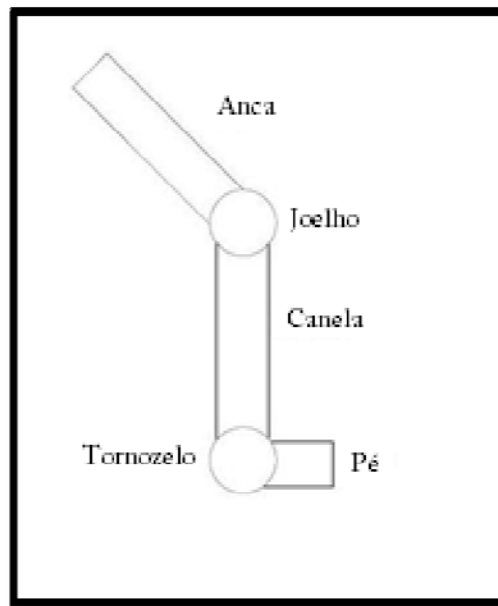


Figura 19 - Representação simples do esqueleto hierárquico de uma perna

As primitivas geométricas que definem a perna estão coordenadas com os ossos a que pertencem. Desta forma, a rotação de uma articulação de nível superior na hierarquia automaticamente desloca os ossos que estão num nível inferior, e portanto as primitivas são também deslocadas em conformidade.

2.4 TÉCNICAS BÁSICAS DE ANIMAÇÃO COMPUTACIONAL

A animação computacional foi buscar vários dos seus princípios à animação tradicional. Neste contexto é importante rever alguns desses princípios bem como outros que foram introduzidos, conforme as técnicas de animação computacional se foram desenvolvendo.

2.4.1 *Keyframing and In-Betweening*

O termo *keyframing* foi utilizado, desde os primórdios da animação, para definir uma técnica de animação. Esta técnica é usada para definir uma sequência animada baseada em momentos chave.

Nos estúdios de animação 2D, os artistas principais desenhavam as frames mais importantes, que, por si só, são suficientes para compreender a animação. Estas frames são chamadas *keyframes* (frames chave). Uma vez estabelecidas e desenhadas as frames chave, os artistas com menos experiência desenhavam as frames intermédias. A esta técnica dá-se o nome de *In-betweening*. *In-betweening* consiste, portanto, na criação da transição de todos os desenhos intermédios que preenchem o

movimento definido pelas *keyframes* [O'Rourke,1998]. Ou seja as frames chave deverão ser suficientes para transmitir integralmente o movimento e as frames intermédias devem ser meras transições entre as frames chave.

Na animação tradicional o processo de criação das frames intermédias é feito através da criação manual de cada desenho intermédio. Na animação computacional, as frames intermédias são normalmente criadas com uma técnica denominada interpolação. Para criar tantas frames intermédias quantas sejam necessárias, podem ser usadas diversas técnicas de interpolação, tais como, a interpolação linear e interpolação curva.

A Figura 20 ilustra o processo de criação de frames intermédias. As frames a preto e a vermelho são as frames chave, enquanto as frames a azul e a verde são as frames intermédias, obtidas por interpolação a partir das primeiras.



Figura 20 - Frames chave e frames intermédias

2.4.2 Interpolação Linear

A interpolação linear é a forma mais simples e a técnica de animação computacional mais directa para calcular frames intermédias. A interpolação linear consiste no cálculo da média dos parâmetros das frames chaves e fornece tantas frames intermédias, com o mesmo espaçamento, quantas as necessárias. Dadas duas poses ou frames chave **Pa** e **Pb**, pode encontrar-se uma pose intermédia **Plerp** através de interpolação linear recorrendo à (Equação 1):

$$P_{lerp} = LERP [Pa, Pb, \alpha] = (1 - \alpha) (Pa) + \alpha(Pb) \quad \text{(Equação 1)}$$

Onde $0 \leq \alpha \leq 1$

Porém, as técnicas de interpolação linear podem produzir resultados pouco naturais quando aplicadas a movimentos subtis, a menos que seja empregada uma quantidade de tempo significativa e habilidade

de animação para afinar os resultados. A interpolação linear é baseada em velocidades constantes entre as frames chave, no entanto isto provoca mudanças abruptas de velocidade nas frames chave onde se passa de uma determinada velocidade (constante) para outra velocidade (constante) diferente. A interpolação linear não se revela útil para manipular mudanças subtis na velocidade, devido ao facto de as frames intermédias serem criadas em intervalos iguais, ao longo da trajectória [Kerlow, 2004].

A Figura 21 representa a interpolação linear sob a forma de gráfico, que tem como parâmetros a distância em função do tempo. No gráfico as frames chave estão representadas por círculos vermelhos e os segmentos de recta pretos representam a interpolação linear entre frames chave. Verifica-se que a velocidade é constante ao longo da interpolação entre duas frames chave.

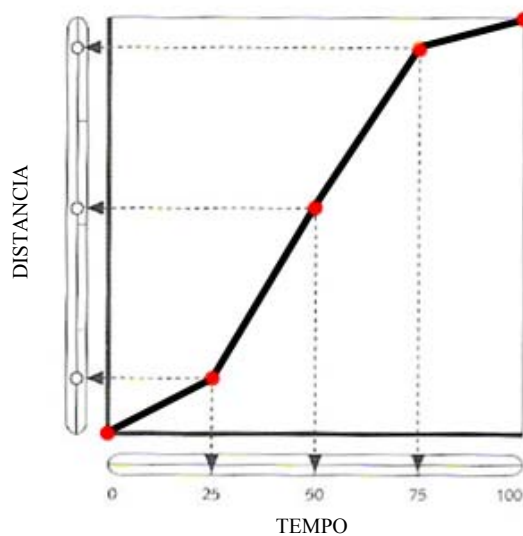


Figura 21 - Interpolação linear, representada sobre a forma de gráfico

2.4.3 Interpolação Curva

A interpolação curva é uma técnica de cálculo de frames intermédias mais sofisticada que a interpolação linear. A interpolação curva faz a média dos parâmetros contidos nas frames chave tendo em conta as variações de velocidade ao longo do tempo (acelerações). A Figura 22 descreve a interpolação curva representada sob a forma de gráfico. As frames estão representadas no gráfico por círculos vermelhos. O aumento da velocidade é representado por uma curva com declive crescente. E o decréscimo da velocidade é representado por uma curva com declive decrescente. Portanto, a distribuição de frames intermédias ao longo da trajectória depende do aumento ou diminuição da taxa de variação. Interpolações curvas podem também incluir movimento com velocidade constante, que é representado com linhas rectas [Kerlow, 2004].

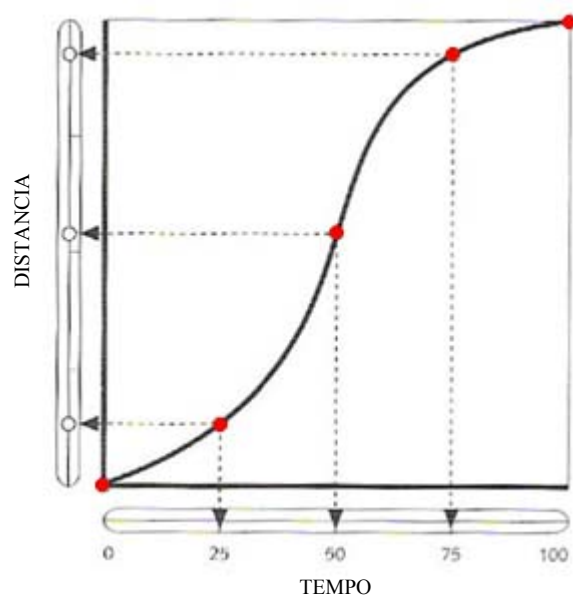


Figura 22 - Interpolação curva

2.4.4 Desenvolvimento do conceito de Actor Virtual

Um actor virtual é um agente autónomo com características especiais. Um actor virtual é um personagem virtual que se assemelha muito mais a um actor real no desempenhar das suas funções. Portanto, é uma analogia ainda sem uma definição nítida e com fronteiras fluidas para outras variações de personagens virtuais. Nesta etapa de investigação e desenvolvimento, e dada a importância previsível que os actores virtuais terão na criação de novos e melhorados sistemas e métodos educacionais, é essencial trabalhar no aprofundamento da compreensão do conceito de actor virtual. Assim, é preciso, nesta fase, identificar as propriedades mais importantes de um sistema de actores virtuais, [Iurgel et al., 2010].

Segundo estes autores o conceito de actor virtual pode ser materializado através da construção de um sistema de animação. Mas é esperado que possam ser dadas outras utilidades aos actores virtuais que estarão relacionadas com este núcleo de animação, portanto, as tecnologias poderão ser reutilizáveis noutros domínios. Por exemplo, se for possível criar um actor virtual apto para a realização de um filme, será certamente possível adaptá-lo para ser um brinquedo de uma casa de bonecas virtual, e modificar os conceitos cinematográficos originais para capacitar uma criança a criar filmes, enquanto brinca, no contexto da casa de bonecas.

Em [Iurgel et al., 2010] foram identificadas, entre outras, as seguintes propriedades específicas de um actor virtual: cinemática através de inteligência artificial; interpolação através de inteligência artificial;

tratamento contínuo de subespecificações; manutenção de coerência horizontal; manutenção de coerência vertical; controlo multi-nível; compreensão da narração; autoria de diálogo.

- **Cinemática através de inteligência artificial:** é um conceito referente à adaptação automática de um actor virtual a mudanças geradas por outro actor.
- **Interpolação através de inteligência artificial:** este conceito é uma analogia à interpolação entre frames chaves através de funções de interpolação. Mas esta interpolação ocorre ao nível de abstracção do comportamento expressivo. O sistema preenche de forma autónoma, comandos em falta relacionados com expressões e reage através da avaliação coerente das implicações, em torno de comandos do autor, e dos parâmetros contextuais.
- **Tratamento contínuo de subespecificações:** o autor pode expedir comandos em vários níveis, mas pode deixar partes da cena subespecificadas, ou especificar, por exemplo, somente comandos a um nível mais baixo, e não especificar comandos de mais alto nível. Isto significa que o sistema completa as falhas de forma autónoma. Se o autor não animar alguma entidade na cena, então o sistema irá completar as partes em falta de forma autónoma.
- **Manutenção de coerência horizontal:** este conceito significa que o utilizador pode definir comportamentos em vários momentos de uma cena. As transições entre comportamentos e a influência mútua dos comportamentos devem ser assegurados pelo sistema. Ou seja o sistema deve garantir a manutenção da coerência dos comportamentos.
- **Manutenção de coerência vertical:** esta manutenção de coerência é necessária quando um comando de nível superior gera comportamentos que podem ser descritos em níveis mais baixos. O sistema deve ser capaz de manter a coerência do desempenho e integrar e completar a interferência do autor.
- **Controlo multi-nível:** os níveis de controlo são organizados hierarquicamente e são os seguintes: nível do corpo virtual; nível motor; nível da expressão comportamental; nível realizador do comportamento.

1. **Nível do corpo virtual:** representa o controlo do corpo do actor, definindo especificamente no tempo, a posição da cabeça, dos olhos, das sobrancelhas, da língua, etc.

2. **Nível motor:** os comandos deste nível podem ser compostos por um conjunto de comandos do nível anterior (corpo virtual). O nível motor difere do nível do corpo virtual, pois é alargado temporalmente, por exemplo permite movimentos como “rodar”, ou “olhar para”.
 3. **Nível da expressão comportamental:** este nível destina-se a expressões que podem ser realizadas, por exemplo, através de uma mudança no olhar, com um sorriso e completado por uma flexão da cabeça. Os elementos do nível da expressão comportamental devem conter, elementos que traduzem individualmente comportamentos do nível motor.
 4. **Nível realizador do comportamento:** este nível destina-se a comportamentos que são hierarquicamente compostos por comandos do nível da expressão comportamental. Onde o “realizador da peça” define orientações específicas para actuação dos actores virtuais. O nível realizador do comportamento contém autonomia de desempenho no sentido que o sistema escolhe a expressão comportamental a empregar para cumprir as instruções.
- **Compreensão da narração:** os actores virtuais devem ser capazes de representar de acordo com a sua interpretação da cena. Isto consiste na análise de diferentes pontos de vista sobre a criação de uma cena, podendo ser a guia para o desenvolvimento de tais faculdades. Permitir que os actores virtuais interpretem de forma autónoma uma história, será determinante para os resultados que podem ser obtidos com actores virtuais.
 - **Autoria de diálogo:** o sistema deverá ser capaz de recolher informações sobre os motivos pelos quais o utilizador criou a cena de uma determinada forma. Isto é importante pois espera-se uma grande ocorrência de informação incompleta.

Os conceitos anteriores devem ser entendidos como directrizes para o desenvolvimento de aplicações interactivas compostas por actores virtuais. Por exemplo, considerando o cenário de uma peça de teatro, pode argumentar-se que a cinemática através de inteligência artificial entre os actores humanos e o actor virtual é necessária para o desempenho, e que a interpolação através de inteligência artificial permitiria a interpolação entre os pontos de viragem da cena, de modo, a criar uma representação coerente, na dependência do desempenho do actor humano. O “realizador da peça” deve ter a capacidade de interferir no desempenho do actor virtual da mesma forma como faria com seres humanos (Controlo de múltiplos níveis). Os actores virtuais podem tomar a iniciativa (Autoria de diálogo) e inquirir o utilizador sobre o que é esperado deles, sempre que a situação não é clara.

Ou seja segundo as propriedades específicas de um actor virtual definidas por [Iurgel et al., 2010], o foco no âmbito desta dissertação será o Controlo multi-nível do comportamento dos personagens virtuais.

2.5 ANIMAÇÃO DE PERSONAGENS VIRTUAIS

Segundo [Reynolds,1999], a animação de personagens pode ser definida como a expressão de emoções ou comportamentos de objectos vivos através da utilização de movimento. O termo *Behavior* tem vários significados: pode significar acções complexas de humanos ou de outros animais baseadas na vontade ou no instinto; poderá significar acções amplamente previsíveis num sistema mecânico simples, ou acções complexas num sistema complexo. Em realidade virtual e aplicações multimédia é por vezes usado como sinónimo para “animação”. Neste caso, o termo *behavior* é usado para referir as acções improvisadas ou acções semelhantes às praticadas por actores reais, mas realizadas por um personagem virtual.

O movimento é um elemento de extrema importância em animações geradas por computador para filmes e jogos. Um personagem altamente detalhado e bem modelado necessita de se movimentar convincentemente [Chilton et al., 2004].

O corpo humano é frequentemente utilizado em animação computacional, sendo um bom exemplo de uma entidade complexa. Num esqueleto humano existem 206 ossos e 360 articulações, tendo todos padrões de movimentos específicos. Para criar uma animação convincente, o animador terá que simular os movimentos de grande parte desses ossos e articulações. Alguns autores consideram que um esqueleto com 14 articulações, tem um “baixo nível de articulação” quando comparado com um esqueleto com 72 articulações, que consideram ter um “elevado nível de articulação”. Algumas destas articulações (por exemplo os cotovelos) têm uma importância crucial, enquanto outras (por exemplo, as vértebras) têm menos importância na animação resultante. A quantidade de articulações consideradas afecta a qualidade da animação resultante, assim como o detalhe dos movimentos gerados.

Para além das articulações e ossos, existem muitos outros factores que determinam a qualidade da animação. Estes incluem propriedades estáticas da imagem tais como a qualidade do modelo 3D e a forma como este é mapeado e renderizado numa imagem.

2.6 CONTROLO DE MOVIMENTO DE PERSONAGENS ANIMADOS

As técnicas mais populares de controlo de movimento de personagens animados são: *forward kinematics*; *inverse kinematics* (IK); *motion capture* (MoCap); *motion dynamics*;

2.6.1 Cinemática directa (*forward kinematics*)

Através da técnica *forward kinematics* ou cinemática directa, os ângulos das articulações são manipulados para obter um movimento específico. A cinemática directa consiste numa estrutura hierárquica, onde cada objecto possui um nível de importância, fazendo com que todos os restantes objectos de nível inferior na hierarquia acompanhem as suas modificações de dimensionamento, translação e rotação. A Figura 23 pretende ilustrar a aproximação da mão do personagem a uma chávena, partindo de uma pose de braços ao longo do corpo (A). Neste caso, primeiro é necessário rodar o ombro (B), depois o cotovelo e a mão (C). Cada rotação aproxima a mão da chávena. Não é possível simplesmente posicionar a mão junto à chávena, pois o braço não consegue seguir o movimento da mão.

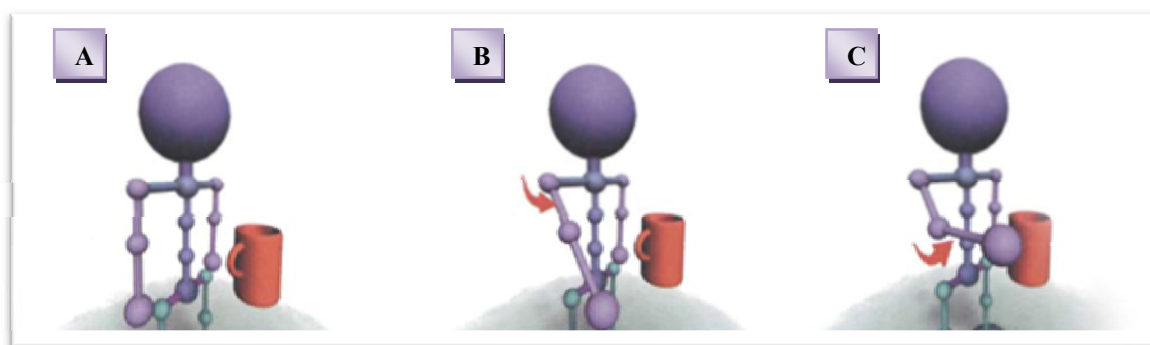


Figura 23 - Cinemática directa [Maestri, 1999]

2.6.2 Cinemática inversa (*inverse kinematics*)

Outra técnica, denominada *inverse kinematics* ou cinemática inversa, tem como característica básica a sua inversão hierárquica. Os elementos de nível hierárquico inferior controlam os elementos de nível superior. Os membros ou objectos são manipulados para a posição pretendida e o software calcula a rotação das articulações que são necessárias para que as posições intermédias sejam criadas. As técnicas de IK são especialmente úteis para animar modelos complexos com grande número de articulações. Um sistema que suporte IK calcula o movimento do esqueleto, sendo necessário especificar apenas os ângulos e as posições de algumas articulações chave, que definem o movimento.

A Figura 24 ilustra três formas diferentes de rodar as articulações do braço de modo que a mão toque a chávena, embora apenas a solução exibida na Figura 24 (A) esteja correcta. Assim, pode acabar por ser necessário indicar à aplicação a forma como as articulações se devem comportar.

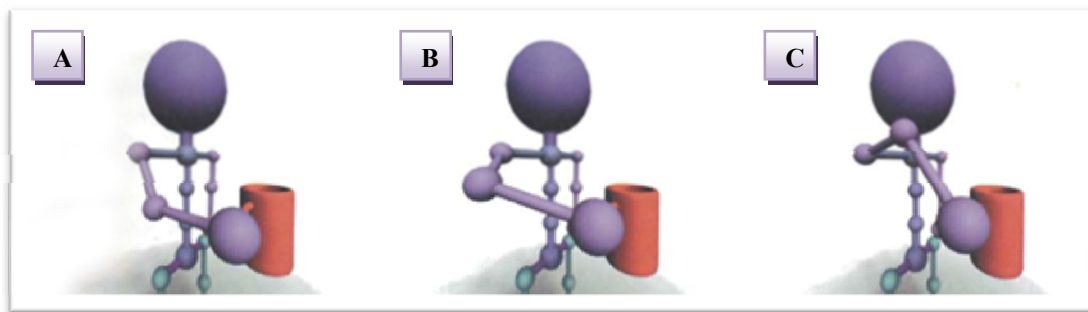


Figura 24 - Cinemática inversa [Maestri, 1999]

2.6.3 Captura de movimentos (*motion capture*)

A técnica *motion capture*, ou MoCap, consiste na aquisição de informação de movimento através da gravação da posição e dos ângulos das articulações de actores reais, ou de objectos em movimento. Esta informação pode depois ser associada a um esqueleto e utilizada para controlar a performance de um personagem virtual. Existem dois tipos de MoCap: *live motion capture*, que é directamente aplicada ao personagem computacional; e *processed motion capture*, onde a informação capturada é refinada e, por vezes, reforçada com a utilização de outras técnicas de animação. Para animar um personagem utilizando a informação capturada é necessário que esta descreva a rotação de todas as articulações e a translação de todas as regiões do corpo. Quando executada com destreza, o MoCap pode trazer para a animação computacional alguns dos movimentos mais naturais de uma performance real [Kerlow, 2004] [Chilton et al., 2004]. Na Figura 25 é possível visualizar a captura da performance de um actor real (lado esquerdo), utilizada para controlar o comportamento de um personagem virtual (lado direito).

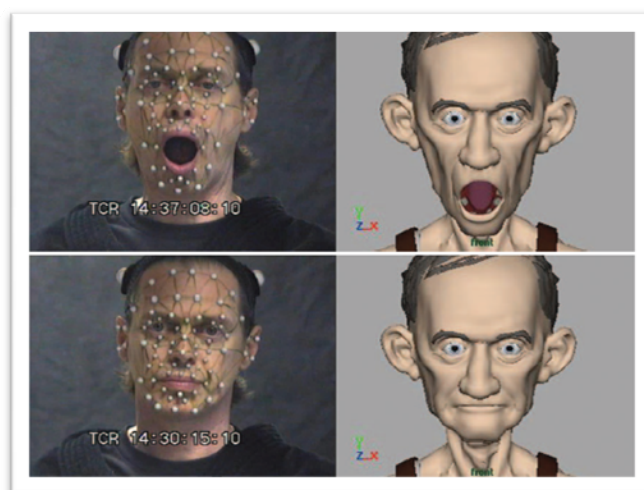


Figura 25 - Captura de performance do actor Steve Buscemi com MoCap para o filme *Monster House* [Whitaker and Halas, 2009]

2.6.4 Técnicas de dinâmica do movimento

Técnicas de dinâmica do movimento (*motion dynamics techniques*) controlam o movimento de objectos tridimensionais através da aplicação de forças às articulações, simulando o movimento resultante no mundo físico. As técnicas de dinâmica do movimento têm em conta variáveis tais como o tamanho dos objectos, o atrito, a gravidade, e outras forças que influenciam o movimento dos objectos [Kerlow, 2004]. Em vez de considerar os modelos humanóides como simples objectos geométricos, estas técnicas pretendem aplicar as leis da física ao modelo através do enriquecimento do mesmo com propriedades físicas do corpo humano. Através de um modelo humano, sujeito a restrições físicas tais como as leis de Newton e a restrições de rotação das articulações, o movimento humano pode ser recriado, poupando a introdução de alguns dados por parte do utilizador. Os métodos físicos puros são mais populares em robótica do que em animação computacional. Pois, apesar de os métodos físicos puros permitirem a criação de movimentos humanóides através da utilização de restrições simples, os movimentos criados são pouco naturais. Desta forma, os métodos físicos puros não são muito utilizados em animação computacional, apesar de estes poderem ser uma ferramenta poderosa quando combinados com outros métodos.

A Figura 26 ilustra a simulação de propriedades físicas de um corpo, para controlo do movimento de um personagem virtual. Este objecto com propriedades físicas é disponibilizado pelo motor de física *Newton Game Dynamics* [NewtonGD]. E é designado por "*Ragdolls*". Consiste num conjunto de articulações especializado que permite aproximar o comportamento do esqueleto humano, sendo essencialmente utilizado para animar sequências onde o personagem “morre”, mas também pode ser utilizado em outros tipos de animações.

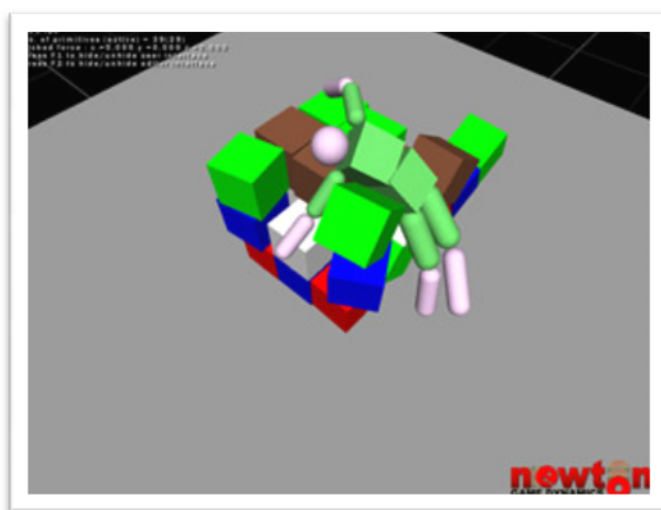


Figura 26 - Técnicas de dinâmica do movimento

2.7 MoCap (captura de movimentos)

No âmbito desta dissertação é importante abordar formas de alcançar animações realistas, alternativas à técnica de animação *keyframe*. A animação *keyframe* consome muito tempo de produção e requer uma grande capacidade técnica, pois assenta em princípios herdados da animação tradicional adaptados para o ambiente computacional, o que implica que mesmo grandes melhorias são traduzidas em ganhos de produtividade muito baixos. A alternativa que se tornou a abordagem padrão para a síntese de movimentos humanos é a técnica MoCap [Giese et al., 2009]. Esta técnica distingue-se por ter uma variedade de aplicações para além da animação, tais como análises biomédicas, vigilância, análise de performances desportivas, ou até como foi apresentado na secção 2.6.3, um mecanismo de controlo e interacção entre computador e humano. Ou seja, apesar de esta técnica ter adquirido bastante importância na animação computacional, muitos dos seus métodos têm as suas raízes em domínios médicos ou biomecânicos.

O termo MoCap refere-se a qualquer método para obter dados que descrevem o movimento do actor a ser capturado. Este actor é mais frequentemente humano, embora animais treinados possam também ser usados. Opcionalmente podem também ser capturados adereços utilizados pelo actor. Os movimentos relevantes são registados e aplicados ao modelo de esqueleto apropriado. Poderão ser feitos ajustes adicionais editando ou misturando os movimentos gravados de forma a modificar o estilo do movimento. Desta abordagem resultam animações com elevado realismo, embora seja necessário um extenso e aborrecido pós-processamento da informação capturada [Giese et al., 2009].

No início do Século XX, Max Fleischer desenvolveu uma técnica de captura de movimento que viria a ser conhecida como *rotoscoping*. Este procedimento está ilustrado na Figura 27 e consiste no decalque de cada frame de um filme, que é projectado na mesa de animação, e que contem a performance de movimentos reais. Esse procedimento facilitou o trabalho dos animadores que começaram a produzir animações bidimensionais com movimentos realistas, que eram impossíveis de alcançar até então. A utilização de *rotoscoping* para a geração de animações 3D era limitada, uma vez que os animadores trabalhavam apenas com imagens 2D e não possuíam nenhuma informação tridimensional [Agarwala et al., 2004].

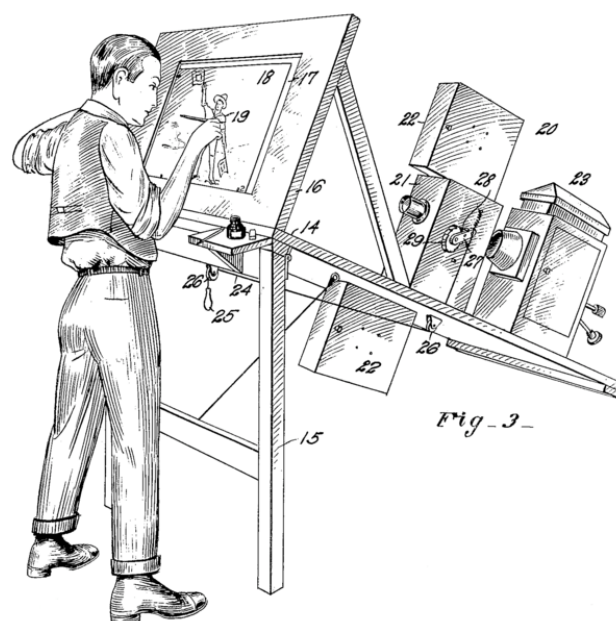


Figura 27 - Exemplo de utilização da técnica *rotoscoping* [Fleisher, 1917]

As técnicas de MoCap começaram a ser praticadas na década de 80, quando foram feitas pesquisas com potenciômetros ligados a actores humanos para medir os ângulos das articulações. LEDs e armaduras mecânicas foram também anteriormente utilizadas para medir a posição e orientação de articulações. A maioria das primeiras aplicações de controlo de movimento estavam limitadas a animar personagens de desenho animado simples ou cabeças e faces, mas não uma animação de corpo inteiro. Actualmente devido à evolução dos sistemas MoCap está a aumentar a variedade de técnicas de animação de personagens baseadas em captura de movimentos [Kerlow, 2004].

A técnica MoCap é diferente da tradicional animação *keyframe*, pois este sistema captura todos os movimentos de actores reais conforme estes se movimentam. Pode ser usada para criar sequências de movimento que podem ser depois enriquecidas com outras técnicas de animação. Os dados que são capturados e guardados como posições das articulações podem ser manipulados directamente e também aplicados a um esqueleto.

Alguns métodos MoCap são mais apropriados para controlo em tempo real de personagens animados, enquanto outros são mais adequados para situações que requerem sequências de movimento complexas com várias camadas de movimento. Excepto quando os dados MoCap são utilizados para controlo de movimento em tempo real, a informação adquirida com o sistema é sujeita a diferentes tipos de tratamentos e afinações no sistema de animação computacional. Isto acontece porque o movimento capturado em bruto normalmente contém demasiado ruído que tem de ser limpo, ou porque este não é suficiente só por si para gerar o movimento requerido [Kerlow, 2004].

O processo de captura de movimentos implica que um ou vários actores reais gerem movimento para um ou vários personagens animados. A preparação dos actores reais e dos personagens animados são duas tarefas, de certa forma independentes: configurar os pontos de amostragem (marcadores) no actor humano e configurar as estruturas hierárquicas que irão controlar o personagem virtual. O posicionamento exacto dos pontos de amostragem depende do tipo de movimento desejado. Mas, em qualquer dos casos, é necessário estabelecer a correspondência entre os pontos de amostragem nos actores reais e as articulações nos personagens animados. Correspondências sem lógica entre os pontos de amostragem e as articulações nos personagens animados podem levar a resultados inesperados. Portanto, é importante verificar se as estruturas hierárquicas de ambos os actores, reais e virtuais, estão estruturadas, para que do movimento capturado resultem os efeitos desejados.

MoCap é frequentemente utilizado para capturar movimentos primários. Os pontos de amostragem são frequentemente distribuídos por toda a cabeça, tronco e membros. Movimentos secundários tais como expressões faciais e gestos são adicionados com outras técnicas de animação.

Os projectos de animação computacional têm uma grande variedade de requisitos em termos de número mínimo de pontos de amostragem e articulações. Não é necessário que as estruturas hierárquicas sejam idênticas, mas quando as hierarquias dos actores real e virtual estão estruturadas de formas diferentes, o movimento resultante não será a transposição directa do movimento capturado. Nestes casos, o movimento resultante será filtrado e modificado. O número de sensores de movimento utilizados numa captura de corpo inteiro varia entre setenta num sistema de elevada performance e doze sensores num sistema com unidades de baixo custo. O posicionamento exacto dos sensores depende de vários factores, tais como o número de sensores disponíveis, o tipo da tecnologia de sensores de movimento utilizada, o tipo de movimento capturado, e o tipo de informação a enviar para o software de animação [Kerlow, 2004].

Para que a informação capturada possa ser utilizada adicionalmente noutros processos computacionais, a informação necessita de ser guardada num formato reconhecível. O formato da informação é normalmente dependente do sistema que está a ser utilizado para capturar o movimento [Chilton et al., 2004]. Um exemplo de formato de informação MoCap é o C3D, que é considerado um formato normalizado. O formato C3D é compatível com sistemas de movimento *Vicon*. As aplicações da *Vicon* como o *Vicon iQ* (ver capítulo 3.1) permitem a leitura e processamento de ficheiros C3D criados por outras aplicações. Outro formato normalizado para animação é o FBX, este formato é amplamente utilizado para a aquisição e troca de informação tridimensional, pois suporta uma ampla variedade de fontes.

A Figura 28 representa o processo de criação de uma animação computacional produzida recorrendo à captura de movimentos. O processo de criação do movimento está dividido nas seguintes etapas:

captura da performance do actor real (A); nuvem de pontos que representa a informação MoCap (B); modelo do personagem com baixa resolução (C); modelo final do personagem (D).

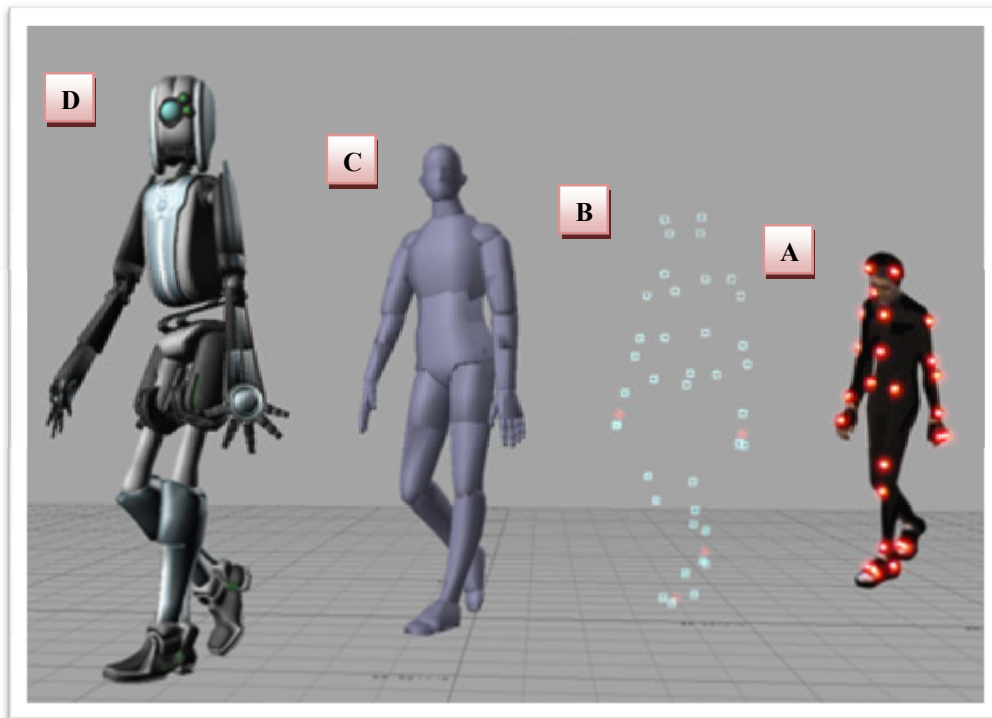


Figura 28 - Animação computacional produzida através de Mocap

A grande vantagem do MoCap é que as sequências de animação são alcançadas de forma relativamente rápida em comparação com a quantidade de tempo dispendido para obter um nível de realismo similar recorrendo a animação *keyframe*. Os pontos fracos do MoCap são a quantidade de informação produzida, e o elevado custo do sistema. Contudo, os desenvolvimentos nas ferramentas de edição e processamento tornam o uso de *Motion Capture* viável para animação [Chilton et al., 2004].

2.7.1 Tipos de Sistemas MoCap

Podemos dividir os sistemas MoCap em três categorias: ópticos, magnéticos e mecânicos. Actualmente, os sistemas mais utilizados são os baseados nas tecnologias óptica e magnética. O princípio por detrás de todos os modos de captura de movimento é essencialmente o mesmo. Um actor é vestido com uma roupa especial para o efeito, realizando depois uma determinada acção. Ao longo da performance dos actores, a posição dos seus membros é normalmente monitorizada, até 250 vezes por segundo e guardada posteriormente num ficheiro. Para avaliar a eficácia do movimento a

informação capturada é aplicada em tempo real a uma versão de baixa resolução do modelo de destino.

2.7.1.1 Sistemas Ópticos

Na Figura 29 está ilustrado um sistema óptico de captura de movimentos cuja base são as câmaras de infravermelhos que são dispositivos fotossensíveis que usam uma matriz de células fotoelétricas para capturar luz, e então medir a intensidade da luz para cada uma das células, criando uma representação digital da imagem. São utilizados marcadores reflectores passivos que são fixos (devido ao velcro da base marcador) a uma roupa especial que o actor veste. O sistema de câmaras de infravermelhos monitoriza a área de captura. Cada câmara tem uma fonte de luz infravermelha (normalmente um anel de luz em volta da lente) que é reflectida de volta pelos marcadores. Uma vez que o sistema esteja calibrado, para saber a posição relativa de cada câmara em relação às restantes, as imagens de cada câmara são usadas para triangular a posição de cada um dos marcadores.

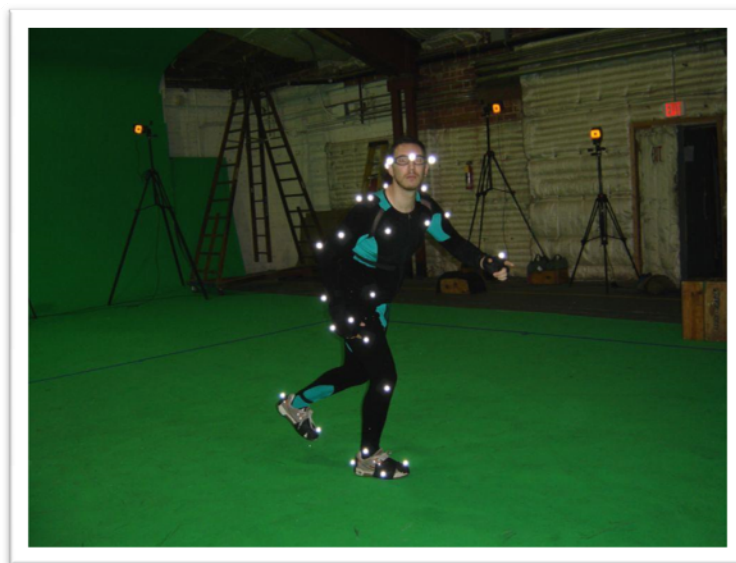


Figura 29 – Sistema MoCap Óptico

Uma das vantagens da utilização de sistemas ópticos é a elevada taxa de amostragem, que permite a captura de movimentos rápidos como os utilizados em artes marciais e desportos olímpicos. A taxa de amostragem depende basicamente da capacidade de definição das câmaras utilizadas no processo. Quanto maior resolução tiverem as câmaras, maior será a taxa de amostragem.

Como os sistemas ópticos dependem da linha de vista, é possível que os marcadores sejam por vezes obstruídos, tornando-se invisíveis (problema de oclusão). Isto pode acontecer, por exemplo, quando o

actor se deita no chão, ou quando é ofuscado por um adereço ou por outro actor [Chilton et al., 2004]. Tal perda de marcadores pode ser compensada através de sistemas de software ou através da diminuição da área de captura. O problema de oclusão pode ainda ser minimizado com a utilização de um número maior de câmaras e de marcadores. Porém existem desvantagens associadas à utilização de um maior número de câmaras, pois este aumento acarreta um maior tempo de processamento para o CPU, durante a monitorização dos marcadores. Ao aumentar-se o número de marcadores surge a dificuldade em identificar os marcadores que estão muito próximos. Este problema é directamente influenciado pela resolução das câmaras. Assim, câmaras de maior resolução conseguem identificar melhor os marcadores próximos.

2.7.1.2 *Sistemas Magnéticos*

Na Figura 30 está ilustrado um sistema magnético de captura de movimentos, onde o actor se posiciona junto de um emissor electromagnético, depois de equipado com sensores posicionados nas articulações do corpo. Os sensores de posicionamento e orientação no campo magnético são gravados e utilizados para especificar a posição e rotação dos membros do corpo. Os sistemas magnéticos são sensitivos, de modo que, objectos metálicos próximos podem causar leituras erradas, e como a área de captura é determinada pelo campo magnético, esta é normalmente pequena [Chilton et al., 2004]. A região de captura é significativamente menor do que a área disponível para a captura óptica. No entanto, uma das principais vantagens da captura magnética em relação à óptica é a inexistência de oclusão. Cada sensor pode ser invocado para fornecer um fluxo ininterrupto de dados de boa qualidade [Lever,2002].

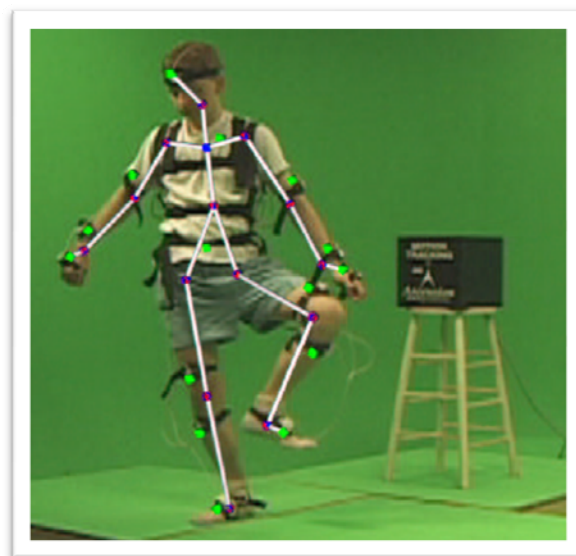


Figura 30 - Sistema MoCap Magnético [O'Brien, 2000]

2.7.1.3 Sistema Mecânico

A Figura 31 ilustra um sistema mecânico de captura de movimentos, que consiste numa estrutura rígida composta por articulações que é colocado sobre o corpo do actor (exoesqueleto). Ao contrário dos sistemas de captura óptica e magnética que determinam a localização de determinados pontos no espaço 3D, o sistema mecânico determina principalmente a orientação. Ao animar um personagem recorrendo à animação *keyframe*, um animador vai passar a maior parte do tempo a efectuar as rotações das articulações. Um sistema mecânico funciona de forma semelhante. Para gravar a rotação numa única direcção é necessário um potenciômetro e uma resistência variável. Variando a resistência obtém-se o efeito de variação da voltagem (diferença de potencial), é utilizado um conversor analógico para digital para converter essa diferença de potencial num valor numérico. Este valor numérico pode ser convertido num valor de orientação para essa articulação. Para permitir que uma articulação se mova em todas as direcções são necessários três potenciômetros para cada articulação, um para cada um dos três eixos de rotação. Um fato completo contém cerca de 50 potenciômetros, que têm que ser monitorizados cerca de 25 vezes por segundo [Lever,2002].



Figura 31 - Sistema MoCap Mecânico

A raiz da hierarquia de um personagem é geralmente a pélvis. Ao contrário das outras partes do esqueleto, a posição e orientação não podem ser calculadas usando um esqueleto rígido externo e potenciômetros. Em vez disso, deve usar-se algum tipo de sensor que calcule a posição exacta e a orientação no espaço. A posição absoluta e a orientação da pélvis são frequentemente calculadas usando um giroscópio digital inercial.

A grande desvantagem da captura mecânica é o peso do fato, que pode restringir a mobilidade do actor.

Uma vantagem é a área de captura disponível. Os sistemas mecânicos, normalmente têm uma área de captura dez vezes maior do que as suas contrapartidas magnéticas e ópticas [Lever,2002]. Outra vantagem importante é os sistemas mecânicos não serem afectados por campos magnéticos ou reflexões indesejadas, problemas típicos de sistemas magnéticos e ópticos. Por isso, não necessitam de um longo processo de calibração, o que torna a sua utilização mais fácil e produtiva.

2.7.1.4 Escolha do Sistema de MoCap mais vantajoso

Quando se decide por uma opção, tem que se obter um equilíbrio entre o custo da captação e as especificações exigências. Um método mecânico em longas distâncias será provavelmente a melhor opção. Se o actor quer sentir-se menos sobrecarregado, então o sistema óptico será o caminho a percorrer. Se é necessária precisão em curtas distâncias, então o sistema magnético irá fornecer os melhores resultados [Lever,2002].

Os sistemas MoCap ópticos possuem um conjunto de propriedades que lhe conferem vantagens e desvantagens relativamente aos outros tipos sistemas MoCap estudados.

As principais vantagens consistem na elevada precisão dos dados ópticos, na maior frequência de captura de que resulta um maior número de amostras por segundo, na possibilidade de utilizar um maior número de marcadores caso seja necessário, ou mesmo mudar as configurações de um marcador (desde que o esqueleto hierárquico do personagem virtual seja adaptado em conformidade). É também possível obter aproximações do esqueleto hierárquico do personagem virtual usando grupos de marcadores (por exemplo, a articulação da cabeça é normalmente definida pelo conjunto de marcadores: frontal esquerdo; frontal direito; posterior esquerdo; posterior direito). Outra vantagem é a grande mobilidade do actor, que não está limitado por cabos nem estruturas rígidas e pesadas. Tem ainda à sua disposição uma maior área de desempenho do que a maioria dos outros sistemas, podendo assim gravar as suas performances com elevada naturalidade.

As principais desvantagens são a necessidade de pós-processamento dos dados ópticos, o elevado custo do hardware (entre 100 000€ e 250 000€) e o facto de as capturas deverem ser realizadas em ambiente controlado, longe de luz amarela e ruído de reflexão. Outra desvantagem é o facto de os sistemas ópticos não poderem capturar movimentos quando os marcadores estão ocultos por um longo período de tempo.

Depois de avaliadas as vantagens e desvantagens dos sistemas ópticos conclui-se que as suas vantagens superam as desvantagens. Destaca-se também a disponibilização por parte do CCG de um sistema óptico de captura de movimentos para utilizar no âmbito desta dissertação, tendo sido um factor decisivo na escolha do tipo de sistema de captura de movimentos.

Quando não é possível recorrer a um sistema de captura de movimentos devido por exemplo aos elevados custos do aluguer de um espaço dotado destes equipamentos, existe sempre a possibilidade de utilizar bibliotecas de movimentos disponíveis na internet. Por exemplo o laboratório de captura de movimentos CMU - *Carnegie Mellon University* possui base de dados que contem milhares de ficheiros de movimento e cuja organização está descrita em [Torre et al., 2009]. A informação MoCap é disponibilizada gratuitamente para fins comerciais e de investigação.

2.8 MANIPULAÇÃO DE INFORMAÇÃO MOCAP

Apesar de ser uma tarefa difícil, MoCap é a melhor forma de criar movimentos humanos realistas. O maior problema do *MoCap* é que o processo de captura precisa de ser repetido para todos os movimentos desejados uma vez que a manipulação de informação *MoCap* não é fácil. Vários investigadores têm trabalhado no problema da manipulação de informação *MoCap* preservando, contudo, o seu realismo. Os seus trabalhos podem ser divididos em duas aproximações: baseados em física e baseados em interpolação. Alguns investigadores usaram a nova informação *MoCap* criada por um destes dois métodos para formarem grafos de movimento, que são estruturas de dados que permitem a criação de movimentos que por sua vez, são combinações de sequências de movimento mais pequenas.

2.8.1 Métodos Baseados em Física

Como foi referido anteriormente (secção 2.6.4) as propriedades físicas dos humanos podem ser usadas para criar movimentos que respeitam as restrições do corpo humano assim como as leis da física. Os movimentos que são criados a partir do zero são normalmente pouco realistas, mas as técnicas baseadas em física podem ser usadas para modificar movimentos existentes de forma a torná-los mais realistas.

A geração de movimento baseado em física usando informação *MoCap* é umas das ferramentas úteis para criar transições entre duas sequências de movimento.

2.8.2 Métodos Baseados em Interpolação

Como vimos anteriormente (secção 2.4.2) a interpolação é uma ferramenta útil para criar transições entre duas frames chave. Quando se faz a interpolação entre duas frames estáticas com velocidade constante, a interpolação resultante pode não parecer realista. Além disso, se as poses não forem suficientemente parecidas, é possível que a interpolação pareça pouco natural. Uma forma de superar este problema é interpolando entre duas sequências MoCap no tempo, em vez de interpolar entre duas frames estáticas.

Para que o movimento resultante seja realista, os movimentos interpolados necessitam de ser suficientemente próximos e a interpolação deve ser curta. Se a interpolação requerida for longa e os tempos dos movimentos interpolados forem diferentes, deve ser utilizado um algoritmo para medir a similaridade entre duas sequências de animação que podem variar no tempo ou na velocidade. A esta técnica, introduzida por [Kovar and Gleicher, 2003], dá-se o nome *Timewarping*. A utilização de *Timewarping* permite alinhar frames semelhantes dos movimentos interpolados. Por exemplo, as semelhanças nos padrões de locomoção seriam detectadas, mesmo se numa animação o personagem estiver a caminhar lentamente e numa outra estiver a caminhar mais rapidamente. A utilização de *Timewarping* reforça a correspondência temporal e reduz o aparecimento de artefactos não naturais.

2.9 COMPOSIÇÃO E TRANSIÇÃO DE ANIMAÇÕES

Em aplicações tais como os jogos, durante o tempo de vida de um personagem, este terá provavelmente que alterar o seu estado de forma a responder a alterações no ambiente. Uma mudança de estado significa, na maior parte das vezes, uma mudança de sequências de animação. Num sistema de animação de personagens a transição entre animações é uma questão importante. Existem vários processos para alternar entre diferentes movimentos (por exemplo alternar entre modo *descanso* e *caminhada*). A ideia mais simplista consiste em fazer as animações de forma a poderem ser concatenadas sem descontinuidades notáveis. Mas o trabalho de preparar as animações desta forma é aborrecido e eventualmente terão de ser criadas transições especiais entre duas animações. Sistemas mais sofisticados contêm mecanismos de mistura de animações ou inserem automaticamente uma fase de transição entre animações [Camilo and Conci, 2006].

A animação em tempo real de personagens 3D pode ser alcançada através da mistura de pequenos excertos de movimentos produzidos por MoCap ou por animação *keyframe*. Os excertos de movimento (clipes) são pequenas animações (por exemplo, *caminhada*; *corrida*; *descanso*).

A mistura de clipes de movimento para criar transições é uma abordagem eficaz em movimentos semelhantes, e uma escolha criteriosa do processo de mistura pode produzir movimentos fisicamente correctos [Safonova and Hodgins, 2005]. Pode ser necessário ajustar o tempo de algumas sequências de animação para criar misturas visualmente agradáveis [Kovar and Gleicher, 2003]. Os autores de [Wang and Bodenheimer, 2004] demonstram que o comprimento de uma interpolação linear é importante. Pode obter-se uma mistura de melhor qualidade misturando mais de duas sequências de animação. Os autores de [Kovar and Gleicher, 2004] descrevem como encontrar vários movimentos que podem ser misturados.

Actualmente o MoCap tem sido uma técnica bem sucedida para criar animações de humanos, especialmente graças aos métodos de edição de movimento. Como a captura está limitada a um conjunto finito de sequências de animação, por si só oferece pouca flexibilidade. No entanto, através da edição de movimento, as capturas podem ser adaptadas para responder a uma situação específica. Alguns dos métodos de edição mais bem sucedidos são baseados na ideia de combinação de movimento (*Motion Blending*) para obter novos movimentos, combinando várias capturas e os respectivos pesos. A combinação de movimentos tem várias aplicações. Por exemplo, pode ser utilizada para criar transições suaves entre animações, permitindo a construção de animações complexas partindo de animações mais simples. Outra aplicação é a interpolação, ou a criação de movimentos intermédios [Kovar and Gleicher, 2003].

A transição entre animações pode ser conseguida através da combinação de animações, tendo como ideia chave a interpolação linear de duas animações. Desta forma durante o período de transição ambas as animações estão activas, a primeira sai de cena gradualmente enquanto a nova animação vai tendo cada vez mais peso na animação resultante. A Figura 32 ilustra o conceito de interpolação usando cores para representar o peso de cada uma das sequências de animação, na animação resultante. Durante o processo de mistura não existem diferenças entre as frames chave e as frames intermédias. Elas são todas frames de uma sequência de animação sendo interpoladas com a frame correspondente da outra sequência de animação [Camilo and Conci, 2006].

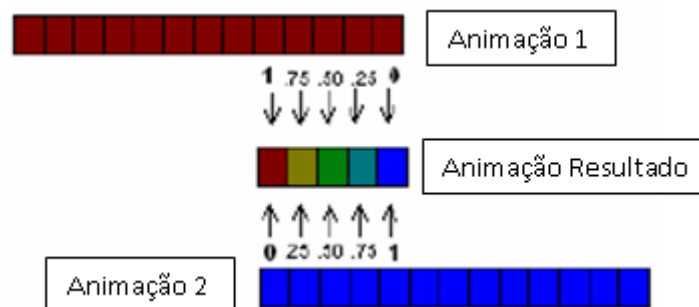


Figura 32 - Transição por composição com interpolação linear

Esta abordagem para a mistura e transição entre sequências de animação não garante que funcione como a transição no mundo real. De qualquer modo pode obter bons resultados, desde que sejam alinhadas sequências similares [Camilo and Conci, 2006].

Existem métodos de mistura sofisticados que podem oferecer vantagens sobre a interpolação linear, mas a criação de movimentos não é habitualmente obtida recorrendo apenas à mistura de animações. A mistura de movimentos pode introduzir um artefacto particularmente indesejado quando os pés do personagem se movem, quando deveriam permanecer imóveis. Este fenómeno é conhecido como *footskate* [Kovar et al., 2002B].

2.10 INTERPOLAÇÃO DE ROTAÇÕES 3D

Os ambientes virtuais são ricos em elementos sujeitos a transformações geométricas cujo cálculo envolve álgebra linear. Neste contexto, o controlo e a animação de objectos gráficos, como por exemplo, personagens virtuais, pode ser obtido através de métodos que permitem efectuar rotações 3D. Nesta secção são abordados os seguintes métodos de interpolação de rotações: ângulos de Euler; matrizes de rotação e quaterniões.

2.10.1 Ângulos de Euler

A representação mais simples para uma rotação 3D é obtida por ângulos de *Euler*. Qualquer rotação 3D pode ser representada com três rotações consecutivas em torno dos eixos coordenados. Deste modo uma rotação R pode ser representada como a composição das três rotações $R1, R2, R3$ (Equação 2) [Kuipers, 2002]:

$$R = R1 * R2 * R3 \quad \text{(Equação 2)}$$

Os eixos de rotação correspondentes a $R1$, $R2$ e $R3$ podem ser escolhidos arbitrariamente. Ao deslocamento em torno dos eixos de rotação X, Y e Z dá-se o nome *roll*, *pitch* e *yaw* respectivamente (Figura 33). Os ângulos (Φ, Θ, Ψ) são utilizados para representar as rotações em torno destes eixos.

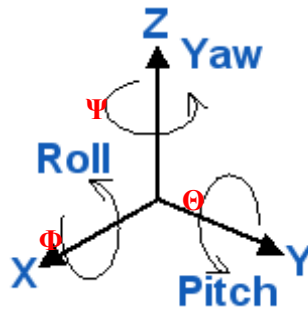


Figura 33 - Sistema de rotação *Pitch-Yaw-Roll*

As rotações de Euler podem ser úteis e são suportadas pelo *Ogre 3D*, mas há uma série de desvantagens na utilização de *ângulos de Euler* para interpolar rotações 3D. A primeira de todas é que a utilização de *ângulos de Euler* tende a levar a problemas de perda de um nível de liberdade. Este efeito indesejável é conhecido por *Gimbal Lock*. O efeito *Gimbal Lock* acontece quando dois dos eixos de rotação estão alinhados. Por exemplo, quando o ângulo θ é 90 graus com uma ordem de rotação **XYZ**, os eixos **X** e **Z** estão alinhados um sobre o outro. Agora as rotações conhecidas como *Pitch* e *Yaw* ocorrem sobre o mesmo eixo, afectando a rotação resultante da mesma forma. Sempre que uma rotação resulta no mapeamento de um eixo sobre outro eixo de rotação é perdido um grau de liberdade.

A designação *Gimbal Lock* resulta da nomenclatura utilizada para definir as componentes de um giroscópio (Figura 34). Os arcos que representam os eixos de um giroscópio designam-se por *Gimbals*. Por isso o nome do efeito é *Gimbal Lock*.

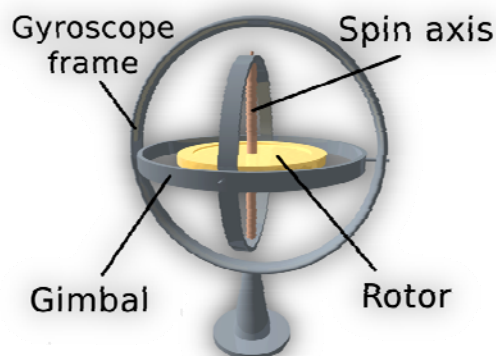


Figura 34 - Esquema de um giroscópio com três eixos de liberdade

A Figura 35 (A) representa a situação normal, em que os três eixos de rotação são independentes. A Figura 35 (B) representa o efeito *Gimbal Lock*, onde dois dos três eixos de rotação estão sobre o mesmo plano.

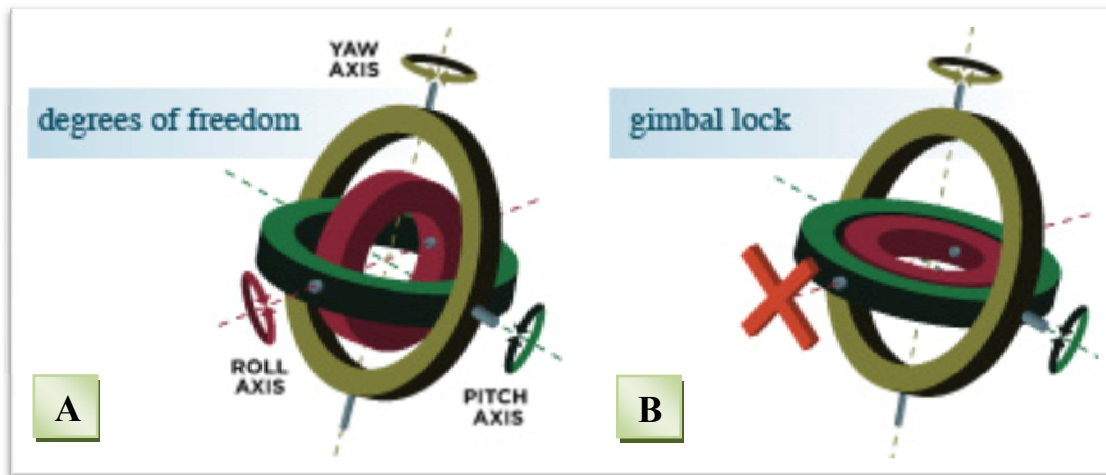


Figura 35 - Situação normal (a), *Gimbal Lock* (b)

Outro grande inconveniente da utilização de *ângulos Euler* para interpolação é que estes são ambíguos. A mesma rotação 3D poderá ser representada por mais do que um valor (Φ, Θ, Ψ) . Se uma das interpolantes verifica essa condição, a interpolação resultante será diferente para valores de interpolação (Φ, Θ, Ψ) diferentes, apesar de estes, representarem exactamente a mesma rotação [Dam, 1998].

2.10.2 Matrizes de Rotação

Como trabalhar com rotações tridimensionais de *Euler* arbitrárias pode produzir uma grande quantidade de operações, as matrizes são geralmente utilizadas para simplificar esse processo. Esta alternativa consiste na utilização de matrizes por forma a representar a interpolação de rotações 3D através da utilização de três matrizes, cada uma para especificar um certo ângulo de deslocamento num determinado eixo X, Y e Z. A rotação sobre cada um dos eixos pode ser armazenada numa matriz individual, por exemplo, *matRotX*, *matRotY* e *matRotZ*. Uma vez armazenados os ângulos de rotação na sua própria matriz, podem combinar-se as operações numa matriz transformação, através da multiplicação das três matrizes (Equação 3):

$$\text{matTransform} = \text{matRotX} * \text{matRotY} * \text{matRotZ} \quad (\text{Equação 3})$$

É necessário ter em conta que a multiplicação de matrizes não é comutativa (a multiplicação da matriz A pela matriz B não é de um modo geral igual à multiplicação da matriz B pela matriz A). Assim como acontece com os ângulos de Euler não é indiferente a ordem pela qual é aplicada cada matriz de rotação. O *Ogre 3D* implementa a classe *Matrix3* para manipular matrizes 3x3, mas é importante ter em conta que a utilização de matrizes para representação de rotações 3D está também sujeita ao efeito indesejado *Gimbal Lock*. De modo a evitar o efeito *Gimbal Lock*, as matrizes podem ser usadas para manipular eixos e ângulos de rotação. Através da definição de um único eixo arbitrário e de um ângulo de rotação é possível obter qualquer rotação [Petkovsek, 2005].

2.10.3 Quaterniões

Quaterniões são construções matemáticas semelhantes a números imaginários, tendo sido introduzidos na Computação Gráfica por Ken Shoemake em 1985. Um quaternião q é composto por um escalar s e um vector tridimensional v e pode ser escrito conforme a (Equação 4):

$$q = (s, v) \quad \text{(Equação 4)}$$

Quaterniões fornecem uma forma directa de representar, ilustrar e calcular rotações 3D. Isto porque podem ser utilizados quaterniões como uma representação alternativa e mais simples que matrizes de rotação. Uma rotação pode ser calculada sem problemas através da multiplicação de um quaternião por outro. A principal razão para a utilização de quaterniões em animação computacional é garantir que a rotação descreve um único e suave movimento no espaço tridimensional [Kremer, 2008].

Uma rotação 3D em torno de um vector n , com um ângulo θ , pode ser representado por um quaternião q do seguinte modo (Equação 5):

$$q = (\cos(\theta/2), n \sin(\theta/2)) \quad \text{(Equação 5)}$$

A forma mais simples de interpolar entre dois pontos é através de interpolação linear (*lerp*). A técnica de interpolação *lerp* é uma fórmula geométrica básica em que, dados o ponto inicial e o final, $p0$ e $p1$ respectivamente, e o parâmetro de interpolação t , $lerp(p0, p1, t)$ cria para cada t um ponto ao longo de uma linha recta entre os pontos $p0$ e $p1$ [Kremer, 2008].

A técnica de interpolação *lerp* é definida pela (Equação 6):

$$lerp(p0, p1, t) = (1-t).p0 + t.p1 \quad \text{(Equação 6)}$$

Contudo, em animação, este tipo de interpolação não é útil, pois as transformações devem ser realizadas de forma suave. Para esse efeito, Ken Shoemake em [Shoemake, 1985] introduziu o conceito

de interpolação linear esférica (*slerp*) associado à representação de uma rotação 3D. A técnica de interpolação *slerp* apenas atinge bons resultados na interpolação entre dois pontos. Por exemplo, os pontos inicial e final de uma rotação simples. Esta técnica é idêntica à técnica *lerp*, diferindo apenas no aspecto em que a interpolação *lerp* é realizada ao longo de uma linha recta e a interpolação *slerp* é realizada sobre a superfície de uma esfera unitária.

A técnica de interpolação *slerp* é formulada pela (Equação 7):

$$\text{Slerp}(\mathbf{q1}, \mathbf{q2}, t) = \frac{\sin((1-t)\theta)}{\sin \theta} \mathbf{q1} + \frac{\sin t\theta}{\sin \theta} \mathbf{q2} \quad (\text{Equação 7})$$

Onde $|\mathbf{q1}| = |\mathbf{q2}| = 1$

A Figura 36 é uma representação bidimensional da interpolação entre quatérniões. O círculo azul é a esfera unitária, os dois vectores amarelos são os quatérniões. O arco vermelho representa o caminho percorrido com a técnica *slerp* e o segmento verde mostra o caminho percorrido com a técnica *lerp*.

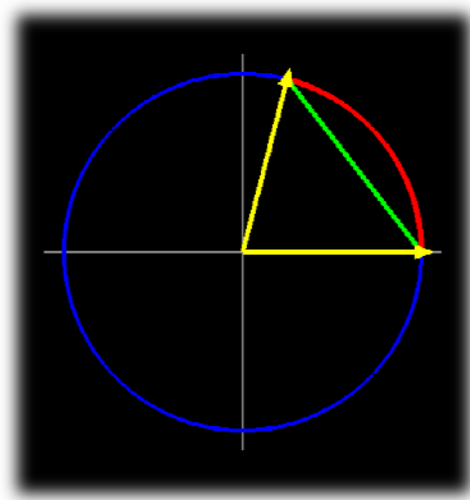


Figura 36 - Interpolação recorrendo às técnicas *lerp* e *slerp*

Outra forma de interpolação entre quatérniões consiste na utilização da técnica *lerp* associada à projecção de cada uma das iterações da interpolação na superfície de uma esfera. Apesar de esta ser a forma mais directa, como se verifica na Figura 37 (A), a interpolação tende a acelerar na região intermédia devido ao aumento das distâncias. A linha vermelha representa a interpolação linear a partir da qual se faz a projecção sobre a superfície da esfera. A Figura 37 (B) representa a técnica *slerp* que é considerada por [Shoemake,1985] como a melhor forma de realizar a interpolação de rotações

3D, pois permite interpolar directamente sobre a superfície esférica, garantindo distâncias uniformes ao longo da interpolação, ou seja a interpolação ocorre com uma velocidade constante.

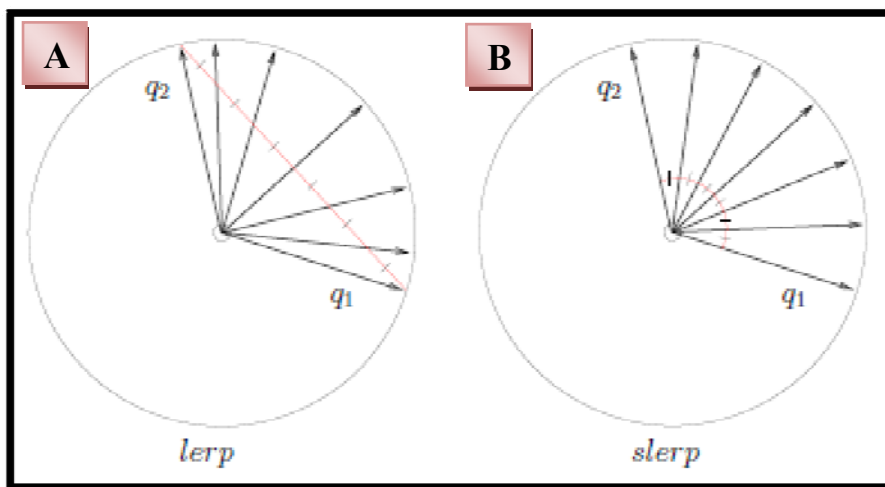


Figura 37 - Comparação entre *lerp* (A) e *slerp* (B).

2.10.3.1 Benefícios da utilização de Quaterniões

Segundo [Petkovsek, 2005], a utilização de quaterniões para interpolar rotações 3D é benéfica em vários aspectos. Um dos benefícios consiste na formação de interpolações bem definidas e únicas. Outra vantagem é que, ao contrário dos ângulos de Euler, a representação de rotações usando quaterniões não sofre o efeito Gimbal Lock.

Através de quaterniões, modificar uma rotação é uma tarefa simples. Por exemplo se tivermos uma rotação que descreve -45 graus, ao criar um novo quaternião que descreve uma rotação de 10 graus e multiplicando os dois, agora temos uma rotação que descreve um ângulo de -35 graus.

Ao rodar um objecto, podemos desejar que este rode suavemente ao longo do tempo. Recorrendo a quaterniões é possível utilizar dois métodos de interpolação: linear (*slerp*) e cúbico (*squad*). O *Ogre 3D* oferece três implementações: *slerp*, *nlerp* e *squad*. A interpolação linear permite especificar a percentagem de interpolação entre os dois quaterniões. Linear significa que a "velocidade" do movimento é constante. *Slerp* e *nlerp* são ambos lineares. *Slerp* é mais preciso, mas mais lento. *Squad* ou interpolação cúbica permite especificar quatro quaterniões **A**, **P**, **Q** e **B**. **P** e **Q** são utilizados para definir uma curva de interpolação de "velocidade" entre os pontos **A** e **B**. Isto permite, numa primeira fase, aumentar a velocidade lentamente, de seguida permanecer com velocidade constante, e finalmente diminuí-la durante a interpolação.

2.11 MÁQUINA DE ESTADOS FINITA

Uma máquina de estados finita (FSM) define as capacidades de movimento de um personagem. Cada estado consiste num conjunto de cliques de movimento que representam comportamentos de alto nível, e cada aresta representa a possibilidade de transição entre dois comportamentos [Lau and Kuffner, 2005]. Num determinado momento cada personagem ocupa um estado e cada um dos estados possui acções ou comportamentos associados. Portanto, enquanto o personagem permanece num determinado estado, ele continua a executar a mesma acção ou comportamento.

Os estados estão ligados entre si através de transições. Cada transição leva de um estado para outro, até ser atingido o estado pretendido (*target state*). Cada um contém um conjunto de condições associadas. Caso sejam alcançadas as condições para desencadear a transição, o personagem inicia a transição que conduz ao estado pretendido. A Figura 38 representa uma FSM simples. Um comportamento pode ser definido como sendo o mesmo que um estado, embora possa existir mais do que um estado que represente o mesmo comportamento. Isto acontece, pois é possível ter múltiplos estados *jog left*, cada um com transições diferentes e conexões diferentes dentro da FSM [Lau and Kuffner, 2005]. Ou seja, numa FSM um comportamento particular pode ser definido através de conjunto de cliques de movimento.

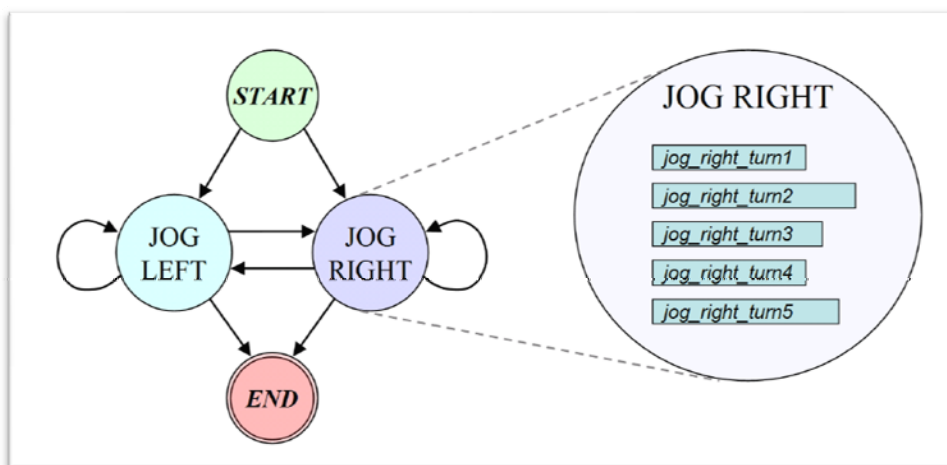


Figura 38 - Máquina de estados finita [Lau and Kuffner, 2005]

2.11.1 Motion Graphs

Na maioria dos jogos, o personagem é controlado interactivamente pelo utilizador, sendo este que decide a próxima acção a tomar pelo personagem. *Disparar* e *saltar* são exemplos de acções que o personagem poderá executar a um dado momento. Alguns jogos possuem dispositivos que permitem a

reação do personagem ao ambiente, o que vai limitar as ações que o personagem pode desempenhar. Existe a assumpção de que se pode fazer a transição entre quaisquer duas sequências de movimento. Esta assumpção apenas é verdadeira se forem verificadas duas condições: a primeira é que o algoritmo de mistura é suficientemente poderoso para oferecer a transição suave, partindo do movimento inicial para o movimento alvo. A segunda condição é que o sistema possa escolher sempre um bom movimento para alternar com o actual. Infelizmente, estas duas condições nem sempre são satisfeitas. A primeira condição está limitada pelos actuais algoritmos de mistura que apenas combinam correctamente movimentos similares, especialmente quando existe uma grande quantidade de ações, ou alterações complexas da pose do personagem. A segunda condição está limitada pela existência de um grafo que contenha as animações disponíveis e as transições entre estas.

Podem ser criadas transições entre sequências de movimento *MoCap*, através de um método baseado em física ou de um método baseado em interpolação. Estas transições permitem, por exemplo, a reprodução da sequência de movimentos A e, usando a interpolação de A para B, passar suavemente a reproduzir a animação B. Esta construção é poderosa, quando aplicada a uma vasta base de dados de sequências de animação. Esta estrutura é conhecida por *Motion Graph* e foi introduzida por [Arikan et al., 2003] e por [Kovar et al., 2002A]. [Arikan et al., 2003] considera os vértices dos grafos como sequências de movimento e [Kovar et al., 2002A] considera as arestas como sequências de movimento. Ambas as aproximações têm as suas vantagens e desvantagens, mas a aproximação formulada por [Arikan et al., 2003] tem sido preferida por mais investigadores, pois é mais intuitiva.

A Figura 39 representa um *Motion Graph* muito simples com apenas duas sequências de animação. Os vértices são as poses e as arestas são as transições entre poses similares. As arestas a preto pertencem à informação original obtida por MoCap e as arestas cinzentas são as arestas adicionais que permitem a transição entre poses similares de sequências de animação diferentes. Qualquer passagem sobre este grafo cria uma cadeia válida de sequências de movimento e a passagem pode ser restringida para fins específicos, de forma a criar propostas que obedeçam às restrições impostas pelo utilizador.

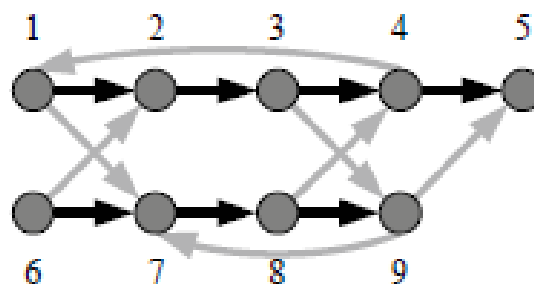


Figura 39- Motion Graph básico

Na indústria dos jogos, estes grafos, eram originalmente criados à mão, sendo neste caso, designados por *Move Trees* (ver secção 2.11.2). Os Investigadores [Kovar et al., 2002A] apresentaram um algoritmo que cria automaticamente o grafo de movimentos, através da detecção de forma automática dos melhores pontos de transição de um determinado vértice para todos os vértices originais, utilizando a medida da distância geométrica. De forma a evitar fins mortos, o grafo é “reduzido”, utilizando apenas as componentes com uma ligação mais forte, ou seja devem ser removidos do grafo todos os vértices e arestas que não são necessários para efectuar a ligação dos vértices originais. Desta forma o grafo reduzido contém a mesma conectividade entre os vértices originais que o grafo original, pois o grafo reduzido contém todos os vértices e arestas do grafo original que são necessários para preservar as melhores transições entre os vértices originais. Depois de criado o grafo, pode pesquisar-se neste uma animação que verifique determinadas restrições. Ou seja, podem ser concatenadas sequências de movimento das quais resulte um movimento mais complexo que permita ao personagem seguir um trajecto específico [Welbergen et al. , 2009].

2.11.2 Move Trees

Uma biblioteca de movimento articulado construída para um personagem virtual é denominada por biblioteca de desempenho. Por outro lado, uma estrutura gráfica construída para definir o fluxo de movimento de um personagem através da biblioteca de desempenho pode ser definida como uma *Move Tree*. As *Move Trees*, tal como os *Motion Graphs*, são estruturas gráficas que representam as ligações na base de dados de movimentos. Porém, as *Move Trees* são criadas manualmente. Além disso, as *Move Trees* são geralmente criadas sem recorrer a um planeamento muito elaborado [Kovar et al., 2002B].

O processo de criação deste tipo de grafo começa na recolha cuidadosa de pequenos cliques de movimento, em sessões de MoCap. Posteriormente são criadas as misturas das animações utilizando ferramentas interactivas.

O controlo em tempo real do personagem é alcançado através do controlo do estado do personagem na *Move Tree*. Ao permitir que o utilizador altere o estado do personagem na *Move Tree*, este obtém o controlo interactivo sobre o personagem.

2.12 SUMÁRIO

A animação por computador constitui uma das maiores linhas de investigação dentro da computação gráfica. Algumas das técnicas de animação computacional, como é o caso da animação *keyframe*, são baseadas em princípios herdados da animação tradicional adaptados para o ambiente computacional, o que implica um elevado tempo de produção e uma grande capacidade técnica dos animadores. Importa por isso encontrar formas de alcançar animações realistas, alternativas à técnica de animação *keyframe*. A alternativa que se tornou a abordagem padrão para a síntese de movimentos humanos é a técnica MoCap, através da qual podem ser registados os movimentos relevantes da performance de um actor real, que uma vez aplicados a um modelo de esqueleto apropriado, permitem animar personagens virtuais com elevado realismo.

A animação de personagens virtuais é rica em elementos sujeitos a transformações geométricas cujo cálculo envolve álgebra linear. O controlo e a animação de personagens virtuais, pode ser obtido através de diversos métodos que permitem efectuar rotações 3D, contudo são os quatérniões que fornecem a forma mais directa de representar e calcular rotações 3D, descrevendo um único e suave movimento no espaço tridimensional.

O planeamento eficiente das restrições dos movimentos é algo a ter em conta num sistema de animação de personagens. Portanto é necessário construir uma estrutura gráfica para definir o fluxo de movimento de um personagem, esta estrutura pode ser definida através de uma *Move Tree*.

3 TECNOLOGIAS

No presente capítulo serão expostas as principais interfaces de desenvolvimento de mundos virtuais, bem como algumas das ferramentas comerciais e de código aberto mais populares neste contexto. O processo de animação de personagens virtuais pode ser levado a cabo recorrendo a diversas ferramentas que vão acompanhando constantemente o aparecimento de novas técnicas de animação. O diagrama da Figura 40 descreve de forma sintetizada as diversas fases do processo de animação computacional adoptado, onde estão descritas as operações realizadas e as tecnologias utilizadas para as realizar.

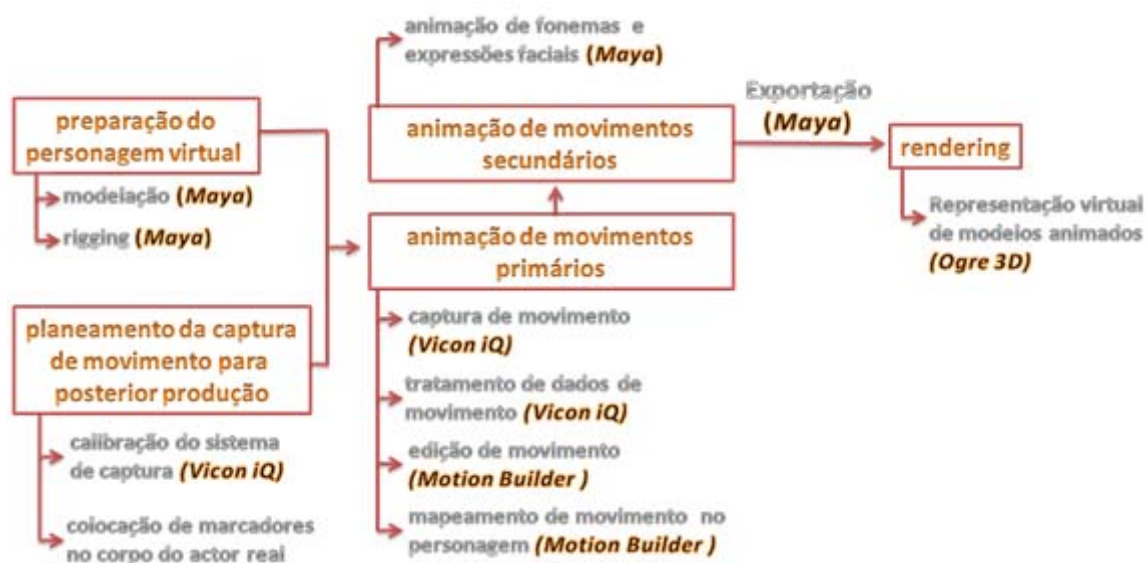


Figura 40 - Processo de animação computacional

3.1 VICON iQ 2.5

A aplicação *Vicon iQ* [Vicon iQ,2004] fornece tudo o que seja necessário em termos de gestão, configuração, captura e pós produção, tudo num ambiente intuitivo e em tempo real. O *Vicon iQ* simplifica drasticamente o fluxo de trabalhos de captura de movimentos e oferece uma elevada precisão quando é monitorizada a interacção entre múltiplos actores, reduzindo significativamente os esforços anteriormente associados à edição manual da informação capturada. O *Vicon iQ* permite o processamento automático mesmo dos cenários mais complicados compostos por múltiplos actores de forma fácil e rápida. O *Vicon iQ* possui algoritmos que resolvem de forma automática a maioria das ambiguidades que estão normalmente associadas aos sistemas ópticos de captura de movimentos através da utilização de um modelo cinemático do actor e respectivos adereços a serem capturados.

Esta aplicação oferece um conjunto abrangente de módulos que podem ser configurados para registar as práticas do utilizador que obtiveram melhores resultados. Ou seja uma vez criado um fluxo de trabalho, este pode ser guardado e posteriormente executado em todas as suas capturas de forma automática [Vicon iQ,2004].

Como se pode verificar na Figura 41 o *Vicon iQ* permite a visualização em tempo real dos dados capturados e da informação 3D dos marcadores associada ao modelo de esqueleto subjacente, indicando o estado do sistema, e permitindo ainda a captura da transmissão em tempo real para processamento imediato e reprodução.

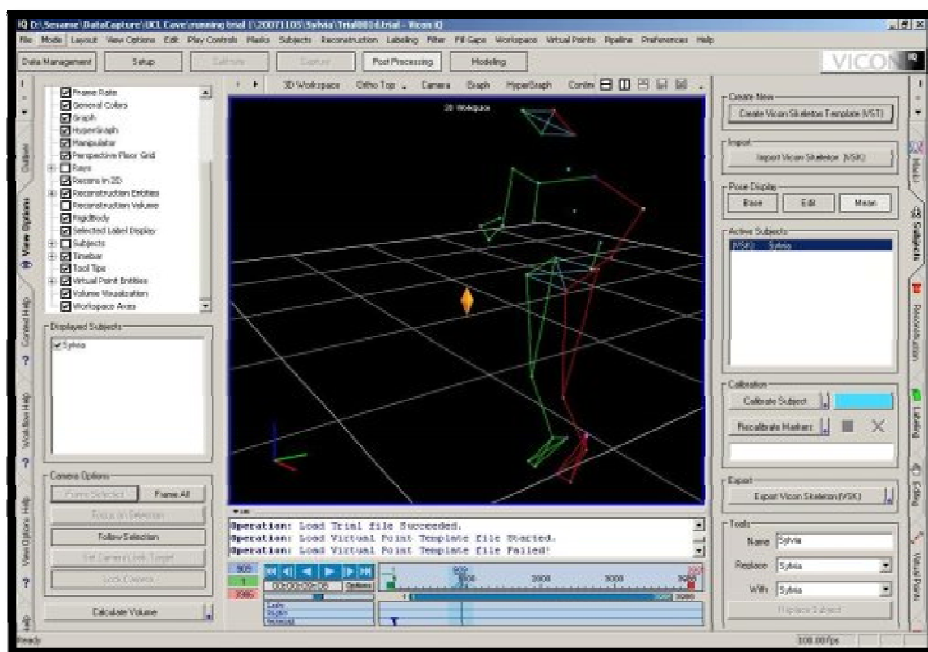


Figura 41 - Análise e edição de informação de movimento utilizando Vicon iQ

Quando o *Vicon iQ* é utilizado em conjugação com as câmaras *Vicon MX*, a precisão e eficiência são bastante elevadas. O *Vicon iQ* utiliza em pleno as capacidades de processamento em tempo real das câmaras *Vicon MX*, permitindo que possam ser detectados marcadores sobrepostos, e centenas de marcadores na cara, corpo e mãos. É também possível capturar simultaneamente múltiplos actores permitindo obter uma verdadeira captura de performance. A característica mais limitadora da aproximação tradicional é que o utilizador só se apercebia de que algo estava a correr mal quando já era demasiado tarde. O *Vicon iQ* elimina este problema através da calibração de um modelo biomecânico do sujeito antes de iniciar a captura. Se surgir uma ambiguidade a qualquer momento, é automaticamente referenciada a informação de uma versão anterior para que o utilizador encontre a melhor solução para completar a captura. A cada passo do processo o *Vicon iQ* mantém o utilizador informado acerca de todas as alterações feitas sobre a informação [Vicon iQ 2.5, 2004].

A Figura 42 representa parte do interface da aplicação *Vicon iQ 2.5* onde pode ser encontrado um painel com seis botões que representam o fluxo de trabalho da esquerda para a direita. As funcionalidades são as seguintes: Gestor de dados; Configuração; Calibração; Captura; Processamento Posterior; Modelação.



Figura 42 - Interface *Vicon iQ 2.5*

Gestor de dados - permite a criação de novos projectos que são organizados em dias de captura, sessões e sub-sessões. Esta secção também disponibiliza informação detalhada sobre cada captura, bem como informação pertinente sobre o estado de cada captura. Para cada projecto é criada uma sessão, rotulada com o dia em que a captura teve lugar

Configuração - permite o ajuste de parâmetros tais como o *Strobe* (*Strobe* é o brilho dos LEDs), ganho (amplificação digital da entrada da câmara), e qualidade do círculo (quão perfeita deve ser a forma para ser reconhecida como um marcador). Esta secção disponibiliza também funcionalidades tais como um monitor de configuração de hardware, que mostra todo o hardware disponível.

Calibração - é a secção que tem mais impacto sobre a qualidade final da informação, é disponibilizado um processo de calibração simplificado, permitindo ainda que o utilizador guarde ou carregue os ficheiros com as configurações da calibração. A calibração é obrigatória sempre que as câmaras são movidas, mas é aconselhável que as câmaras sejam recalibradas a cada sessão de captura.

Captura - é a secção na qual é realizada a captura de movimentos, é também nesta fase que é possível criar objectos que iram facilitar o processo de rotulagem dos marcadores. Nesta fase é aconselhável fazer-se o máximo de capturas possível, antes de se iniciar a fase de processamento, pois não seria correcto fazer esperar o(s) actor(es) pelo demorado processo de limpeza da informação.

Processamento Posterior - permite a reconstrução da informação, rotulagem e edição da trajectória, bem como a criação e edição de sujeitos, tendo como base os modelos de esqueletos *Vicon*.

Modelação - é uma etapa não inerente ao processo de captura de movimento. Inclui a funcionalidade de criação de objectos 3D rígidos que podem ser utilizados para apresentar alguma informação em falta aos sujeitos na etapa “Processamento Posterior” e permite também a localização de marcadores relativamente a marcadores adjacentes.

Apesar de ter sido a ferramenta utilizada para capturar e processar a informação de movimento, o *Vicon iQ* é uma ferramenta actualmente descontinuada. Neste momento, a *Vicon* recomenda a ferramenta *Vicon Nexus*, referindo que esta aplicação permite maior controlo sobre a informação, permitindo novos parâmetros de reconstrução. Isso é particularmente útil em ambientes mais arrojados, permitindo obter dados de melhor qualidade.

3.2 AUTODESK MOTIONBUILDER

O *Autodesk MotionBuilder* é um software de animação de personagem 3D que oferece uma vasta colecção de ferramentas de animação em tempo-real, realizando desde as tradicionais animações *keyframe* às complexas edições de *Motion Capture*. Permite a criação eficiente de animações para jogos ou filmes. Possibilita a criação, a edição e a reprodução de personagens animados complexos, num ambiente interactivo que é ideal para grandes volumes de animação, tais como, cinematografia virtual, pré-visualização e animação de desempenho (MoCap) [MotionBuilder].

O *MotionBuilder* é construído sobre uma arquitectura de exibição em tempo-real, que permite trabalhar sem a necessidade de perder tempo a gerar pré-visualizações das animações produzidas, pois o que se visualiza na “cena” será o mesmo que o resultado final. Para permitir a produção de animações de qualidade é disponibilizada uma lista repleta de ferramentas, para criar animações *keyframe* avançadas e animações em camadas. O manuseamento de informação MoCap ilustrado na Figura 43 é possível graças à importação de ficheiros C3D que contêm a informação de movimento proveniente da aplicação *Vicon iQ*. O manuseamento da informação é simplificado através de ferramentas integradas para editar e misturar animações. Possui ainda tecnologia para a extracção de

fonemas o que permite gerar animações faciais realistas e a sincronização dos lábios independentemente da linguagem utilizada.

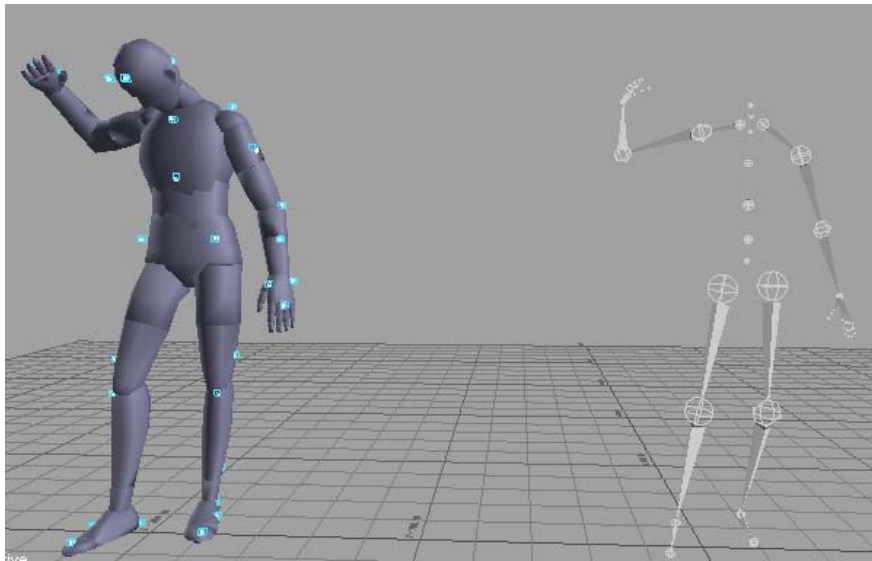


Figura 43 - manuseamento de informação MoCap

As funcionalidades do *MotionBuilder* podem ser expandidas facilmente através de scripts na linguagem Python, ou através do SDK da ferramenta, o *Open Reality*, baseado em C++.

O *MotionBuilder* é facilmente integrável com os ambientes de produção existentes pois é compatível com sistemas de gestão de produção e motores de *render* baseados em hardware. Tudo graças ao suporte nativo do formato FBX, garantindo ainda a transferência de objectos do *3ds Max*, *Maya*, ou qualquer outro software 3D que suporte este formato.

A *StoryTimeline* do *MotionBuilder* oferece um editor não linear integrado no software, que permite a combinação em tempo real de som, vídeo, elementos 3D e câmaras. Faculta assim a capacidade para montar uma “história”, ou seja permite a criação de uma animação complexa, tendo como ponto de partida um conjunto de sequências de animação de complexidade inferior. A Figura 44 ilustra a criação de uma “história”, que consiste num conjunto de sequências de animação. Ou seja possibilita a composição de um conjunto de animações de esqueletos inicialmente contidas em ficheiros FBX independentes num único ficheiro FBX, o que facilita a criação de uma biblioteca de acções que o personagem conseguirá desempenhar.

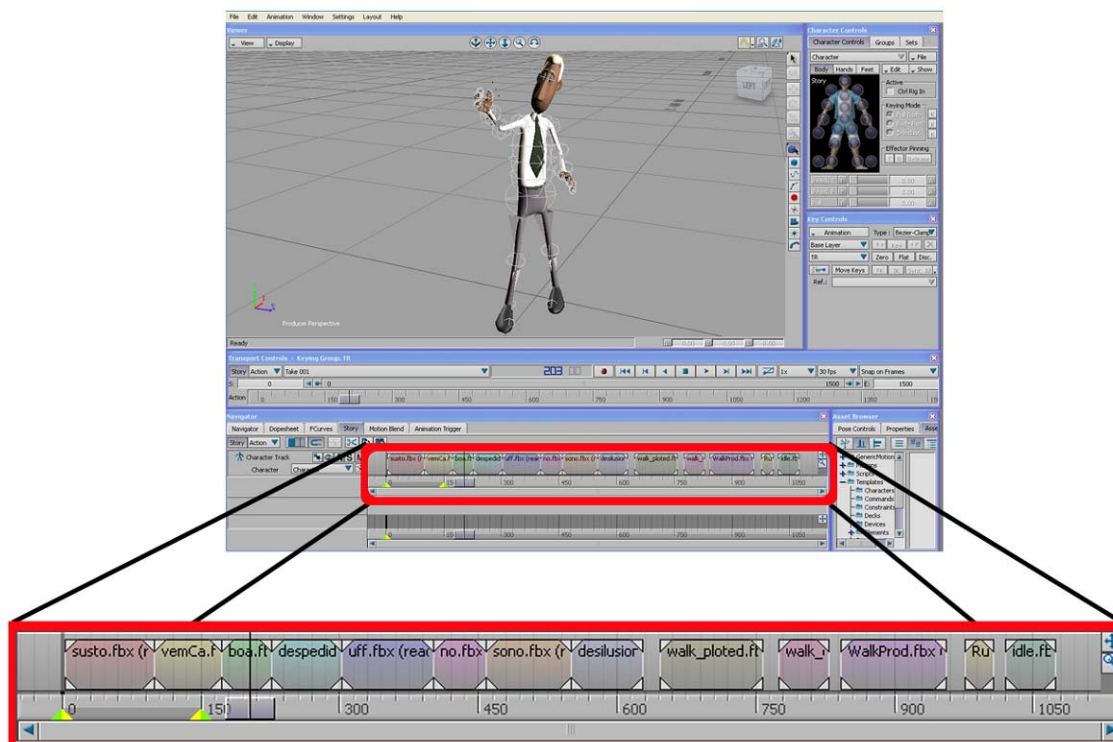


Figura 44 - Criação de uma "história" em *MotionBuilder*

3.3 AUTODESK MAYA 2009

O software *Autodesk Maya* oferece aos artistas um fluxo de trabalho criativo do início ao fim do projecto, integrando ferramentas abrangentes de animação, modelação, simulação, efeitos visuais, *rendering* e composição 3D, tudo numa plataforma de produção altamente extensível. Todas estas funcionalidades estão reunidas numa única aplicação, proporcionando uma mais-valia aos artistas gráficos em computador [Maya].

O *Maya* é um software extenso com diversas funcionalidades. No entanto, está organizado para conter o máximo de informação possível numa interface que se adapta às funções que estão a ser desempenhadas num dado momento. Tendo isto em conta, a interface do *Maya* é organizada para conter muita informação, da maneira mais prática e simples possível. Sendo assim, possível configurar o interface da maneira mais adequada ao tipo de trabalho a desempenhar, seja de modelação, animação, efeitos dinâmicos, etc.

Funcionalidades interessantes no âmbito desta dissertação:

- Modelação de personagens virtuais
- Criação de esqueletos hierárquicos

- *Rigging* de personagens virtuais
- Animação por *keyframe* e animações procedimentais.
- Animação de expressões faciais.
- Animação baseada em poses.
- Suporte nativo FBX, que permite a interoperabilidade entre o *Maya* e o *MotionBuilder*
- *Plugin* de exportação, de malhas e esqueletos para *Ogre 3D*

3.3.1 *Rigging* de personagens virtuais

Uma vez modelado, é necessário preparar o personagem para a animação. Este processo é conhecido como *rigging*. O objectivo deste processo é adicionar um esqueleto e controlos ao modelo de forma que um animador possa manipular e animar o personagem. Um esqueleto construído adequadamente pode ser manipulado de forma simples e rápida, permitindo atingir qualquer pose. No entanto um esqueleto mal configurado pode criar mais problemas do que aqueles que resolve. Uma vez construído o esqueleto, este pode “deformar” o personagem de forma que, idealmente, o personagem virtual pareça vivo aos olhos dos observadores. Para adaptar o esqueleto ao modelo é necessário compreender o processo de animação e usar esse conhecimento para obter uma configuração eficiente. Na Figura 45 é possível ilustrar o processo de associação de um esqueleto a um modelo. Quando esta configuração é perfeita permite que o animador obtenha um maior controlo sobre o personagem, enquanto manipula algumas regiões do personagem de forma automática sem ter que pensar nisso.

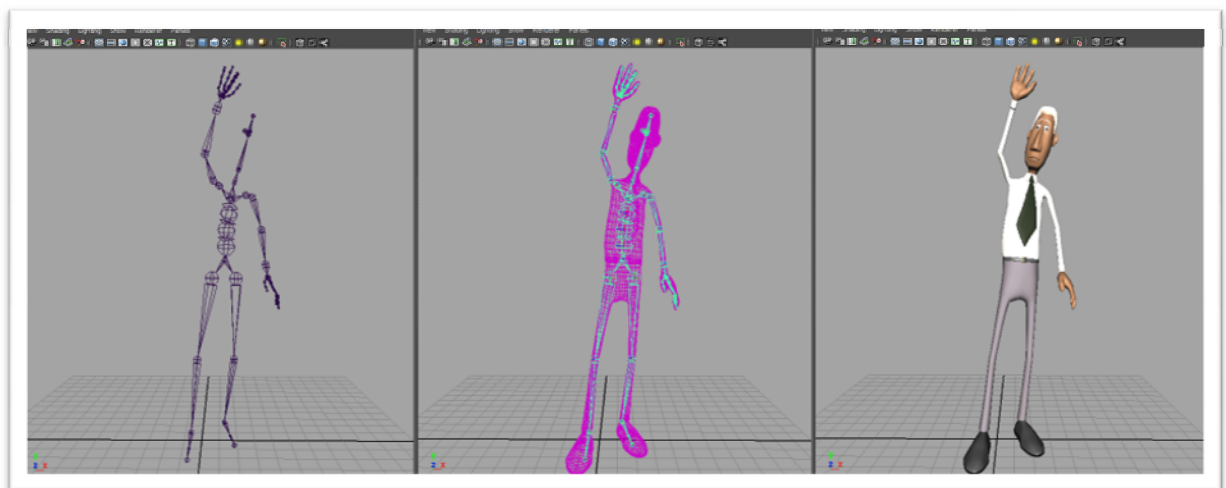


Figura 45 - *Rigging* de personagens virtuais em *Maya* 2011

3.3.2 Animação de expressões faciais

A produção de animações de expressão facial pode ser atingida utilizando algumas opções existentes em praticamente todos os softwares de produção de conteúdos 3D. Na maioria deles está disponível uma ferramenta que grava a posição dos vértices, permitindo a interpolação dessas posições. Isto permite, por exemplo, obter animações do movimento dos lábios ou das maçãs do rosto do personagem. No software de produção de conteúdos 3D *Blender* esta ferramenta é conhecida como *Shape Keys*. Já no *3dsMax* existe a ferramenta *Morph Target* e no caso do *Maya*, esta ferramenta é conhecida por *BlendShapes*. Apesar dos nomes diferentes, estas ferramentas compartilham exactamente a mesma função, que está ilustrada na Figura 46, e que consiste na criação de expressões do rosto, para que o personagem possa expressar momentos de *raiva*, *felicidade*, *tristeza*, *dúvida*, entre outros. Com estas ferramentas podem também ser criados fonemas, que podem ser utilizados para animar um diálogo entre personagens.

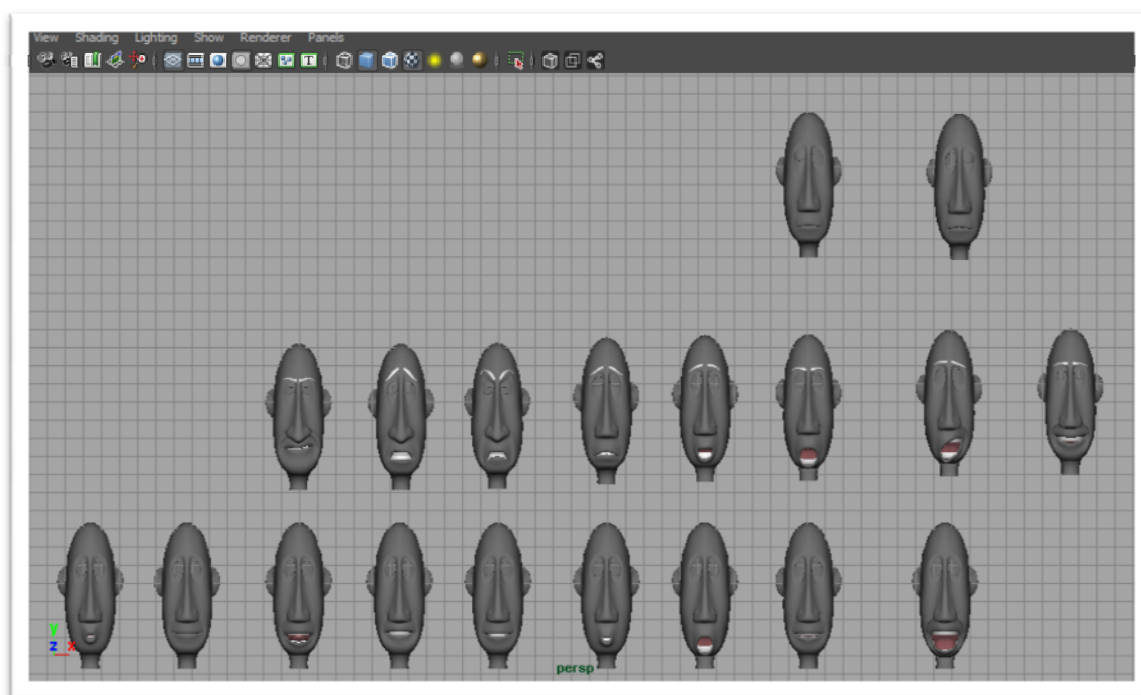


Figura 46 - Animação de expressões faciais

Como se pode verificar na Figura 47, depois de construídas as expressões faciais e os fonemas, basta determinar a percentagem de influência que cada delas deve exercer na obtenção da expressão facial

do personagem. Neste caso a expressão facial foi obtida através da influência das sobrancelhas na pose *alegre* (zona amarela), e da face em pose *alegre* (zona vermelha).

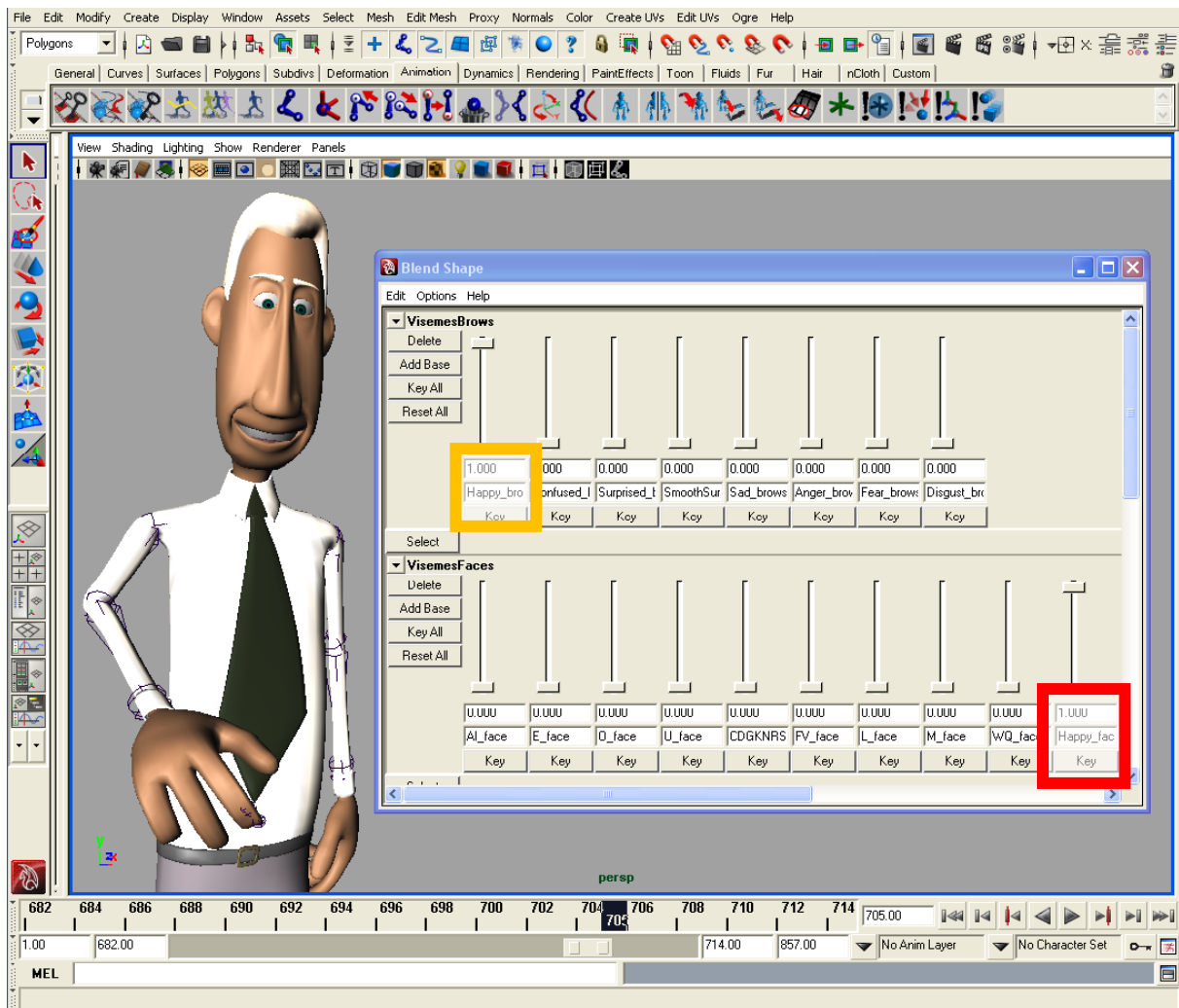


Figura 47 – Configuração de blend shapes

3.3.3 Exportação, de modelos e animações para *Ogre 3D*

Para fazer a exportação para *Ogre 3D* das malhas animadas no *Maya* podem ser utilizados vários *plugins*, como por exemplo o *OgreMax Scene Exporter*, *Maya Exporter* e *LFA Scene Manager*. Dos *plugins* testados o *Maya Exporter* obteve melhores resultados. O *Maya Exporter* é um *plugin* que pode ser instalado sobre o *Maya*, desde que as versões do *plugin* e do *Maya* sejam compatíveis, permitindo a criação de ficheiros nos formatos que o *Ogre 3D* possa carregar. Assim é possível a exportação das malhas dos objectos, com múltiplos materiais e respectivos esqueletos animados. Após o processo de exportação, que esta representado na Figura 48 são criados os seguintes ficheiros: MESH; SKELETON; MATERIAL.

- **MESH** - é um ficheiro binário (convertível para XML) que define a malha. Cada ficheiro MESH referencia internamente o ficheiro SKELETON correspondente, de modo que renomear o ficheiro da malha ou o do esqueleto implica que estes deixem de funcionar como um conjunto.
- **SKELETON** - é um ficheiro binário (convertível para XML) que contem o esqueleto e a informação de movimento. Os ficheiros SKELETON geralmente são colocados na mesma pasta que contem os ficheiros das malhas (\OgreSDK\media\models).
- **MATERIAL** - é um ficheiro de texto que especifica a localização das texturas e parâmetros dos materiais. Estes ficheiros podem necessitar de alguns ajustes pontuais por forma a obter-se o aspecto idêntico ao original. Os materiais do *Ogre 3D* devem ter nomes exclusivos, de modo que o *Ogre 3D* possa distingui-los. Estes ficheiros são geralmente colocados na pasta \OgreSDK\media\scripts.

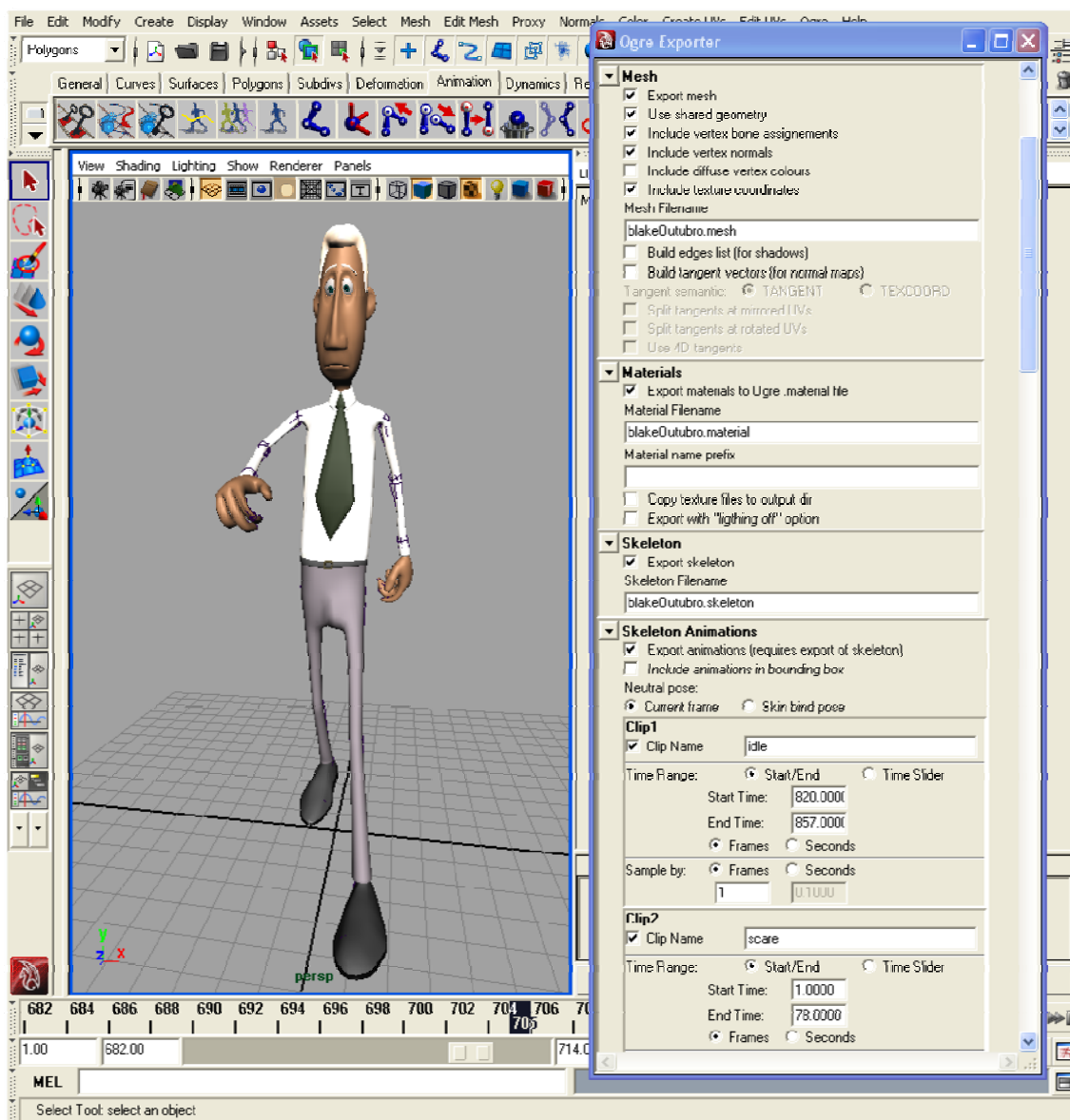


Figura 48 – Exportação de modelos

3.4 *Ogre 3D* – MOTOR GRÁFICO 3D

O *Ogre 3D* (*Object Oriented Graphics Rendering Engine*) é um motor de *render* em tempo real de código aberto. Ou seja, é uma plataforma gráfica que disponibiliza um complexo ambiente gráfico tridimensional, reduzindo e simplificando o difícil trabalho de operar neste ambiente. A qualidade das imagens geradas é muito boa e o processo de *rendering* é bastante rápido. O *Ogre 3D* conta com uma extensa comunidade de utilizadores e está em constante desenvolvimento, é mantido por uma pequena equipa de desenvolvimento que trabalha no seu núcleo principal, e também tem recebido o contributo de muitos utilizadores via internet.

O *Ogre 3D* é um sistema multi-plataforma, podendo ser utilizado com Linux, Windows e Mac OS X. Também existe a possibilidade de escolha entre os dois renderizadores do mercado, o OpenGL e Direct3D. Apesar de a linguagem de programação padrão do *Ogre 3D* ser o C++, é possível utilizar outras linguagens, como Java, Python e C# [Junker, 2006].

Permite a exportação de modelos 3D a partir de várias ferramentas de modelação incluindo *3dsMax*, *Maya* e *Blender*. Outro aspecto importante é o suporte de animações de esqueleto, permitindo a mistura de várias animações.

É importante referir que o *Ogre 3D* não é uma plataforma específica para criar jogos e tão pouco um motor de jogo, pois não trata detecção de colisões, simulação de fenómenos físicos e tratamento de som. O *Ogre 3D* é apenas um componente de um todo, que tem como finalidade somente a função de ferramenta de renderização gráfica. Com o *Ogre 3D* é possível realizar, sem limitações, qualquer tipo de aplicação que necessite de um ambiente gráfico tridimensional, seja um simulador de voo, jogos na primeira pessoa, jogos na terceira pessoa, ou software de visualização CAD. Não importa o fim que terá a aplicação, o *Ogre 3D* disponibilizará o ambiente gráfico, sem restrição quanto ao desenvolvimento.

O OGRE.MESH é o formato nativo de suporte de malhas do *Ogre 3D*. É possível fazer a conversão dos ficheiros OGRE.MESH para OGRE.XML e de OGRE.XML para OGRE.MESH, o que se revela particularmente útil na edição pontual da geometria através do ficheiro XML, assim evita-se o recurso a uma ferramenta de modelação 3D para proceder a uma pequena alteração da geometria, evitando-se também um novo processo de exportação. À semelhança dos ficheiros OGRE.MESH também os ficheiros OGRE.SKELETON são editáveis através de XML.

3.4.1 Skeleton

Em *Ogre 3D* um esqueleto define a hierarquia parental. É conhecido que “O osso do pé esta ligado à ante perna, que por sua vez esta ligada ao joelho” e assim sucessivamente. Tipicamente, todos os ossos que compõem um esqueleto terão pais excepto um, que é o osso *root* (origem). A hierarquia parental é utilizada para propagar transformações ao longo da hierarquia pelos ossos filhos. Por exemplo, se quisermos mover o osso da anca, os ossos filhos da anca (por outras palavras, o resto da perna) terão que ajustar a sua posição e rotação em conformidade. Isto é uma analogia perfeita com o esqueleto humano.

Como foi mencionado anteriormente, as técnicas de IK facilitam bastante o processo de animação de um esqueleto hierárquico. De qualquer forma, o *Ogre 3D* não vem dotado de suporte IK, ou seja, quando um determinado osso se movimenta, o *Ogre 3D* não actualizará as posições dos seus ossos pais. Desta forma, para recorrer a IK para criar animações, é necessário utilizar ferramentas de modelação 3D com suporte IK para criar as *keyframes* que o *Ogre 3D* possa usar e interpolar [Junker, 2006].

3.4.2 Animation State

A primeira interacção entre a aplicação e uma animação em *Ogre 3D* é através de um *animation state*. Nas ferramentas de modelação 3D (*3dsMax*; *Maya*; *Blender*), podem ser definidas múltiplas animações para um esqueleto ou malha através da linha de tempo. Como parte do processo de exportação, podem ser dados nomes às zonas distintas da linha de tempo da animação de um objecto. Esses nomes serão utilizados no *Ogre 3D* para endereçar as diferentes animações de uma determinada entidade. O objecto retornado é um *animation state* que fornece o acesso às seguintes propriedades da animação:

- *Length*: obtém, a duração, em segundos, de um segmento de animação.
- *Current position*: atribui ou obtém a posição actual, em tempo decorrido (em segundos) desde o início do segmento de animação (não desde o início de todas as animações).
- *Animation name*: desde que esteja garantido o acesso à lista de todas as animações de um objecto, isto fornece acesso apenas em modo de leitura ao nome da animação.
- *Looping*: atribui ou obtém se a animação irá ou não ser repetida quando for atingido o final do segmento de animação.
- *Enabled*: atribui ou obtém se a animação está ou não activa.

- *Weight*: apesar de existirem algumas restrições as animações podem ser misturadas. Esta propriedade atribui ou obtém a quantidade de influência que esta animação exerce quando misturada com outras animações.
Os pesos da mistura de animações são processados de forma diferente dependendo do tipo de animação que está a ser misturada.
- *Blend weight averaging*: Quando no *Ogre 3D* é escolhido o modo de mistura de animações obtidas através da média dos pesos (*ANIMBLEND_AVERAGE*), a soma desses pesos deverá ser sempre 1.0. Se a soma de todos os pesos não for 1.0, então o *Ogre 3D* fará a normalização de todos os pesos de forma que a sua soma seja 1.0. Este tipo de mistura está disponível apenas para as animações de esqueleto, sendo neste caso o tipo de mistura por omissão.
- *Blend weight accumulation*: Quando se utiliza o modo de mistura *ANIMBLEND_CUMULATIVE*, o *Ogre 3D* simplesmente acumulará os efeitos de todos os pesos referenciados, sem reequilíbrio. Este tipo de mistura é o único tipo de mistura disponível para animação de vértices (*pose* e *morph*), estando também disponível para animações de esqueleto.

O *animation state* é também onde se actualiza a animação para que esta avance no tempo. Este tempo é sempre o tempo decorrido desde a última chamada para avançar na animação [Junker, 2006].

3.4.3 Tipos de animação suportadas pelo *Ogre 3D*

O *Ogre 3D* suporta a animação por esqueleto, através do ficheiro *OGRE.SKELETON*. Permite também a combinação de várias animações com pesos variáveis. Possibilita ainda o controlo manual das articulações. Para além do suporte de animação por esqueleto, são ainda suportadas animações de formas flexíveis ao nível dos vértices, conhecidas como *Morph animation* e *Pose animation*. Como estas animações são manipulações directas dos vértices, estão contidas nos ficheiros *OGRE.MESH*.

3.4.3.1 *Skeletal Animation*

Skeletal animation é o tipo de animação que o *Ogre 3D* disponibiliza para animações contidas num esqueleto, sendo o tipo de animação de utilização mais frequente. Com a animação de esqueleto, os vértices de uma malha estão associados a ossos que compõem um esqueleto (tal como acontece no corpo humano), esta associação dos ossos à malha (*rigging*) é normalmente obtida numa ferramenta de modelação 3D, neste caso utilizou-se o *Autodesk Maya*. A diferença entre o corpo humano e a animação computacional é que os ossos em animação computacional não existem na realidade, estes

são representados por transformações que definem a posição, a orientação e a dimensão do osso. Se olharmos para os dados da exportação de um esqueleto, não encontraremos referência à geometria.

As animações de esqueleto operam através do cálculo da posição dos vértices como função da matriz transformação que representa cada osso. As frames chave numa animação de esqueleto são simplesmente deslocamentos relativos das restantes posições, orientações e dimensões de cada osso. Estas são combinadas para formar os ossos num determinado ponto ao longo da linha de tempo da animação [Junker, 2006].

3.4.3.2 *Vertex Animation*

Vertex animation consiste, em animar directamente a posição actual dos vértices em vez de animar os ossos de um esqueleto e calcular as posições dos vértices em função desses ossos. A animação de vértices requer bastantes recursos (utilização de várias malhas), já que envolve o transporte em torno de uma cópia completa da informação de um vértice animado para cada *morph target* ou *pose* da animação. De qualquer forma, por vezes é a única forma de obter um efeito particular, como por exemplo, uma expressão facial convincente.

As animações de vértices são geridas ao nível da sequência de animação. Isto significa que podem ser animadas diferentes regiões de uma malha recorrendo a diferentes tipos de animações de vértices. Por exemplo, pode animar-se a cabeça e face de um modelo utilizando o método mais complexo mas também mais flexível que é o *pose animation*, e animar o resto da malha do modelo utilizando um método mais simples mas também menos flexível que é o *morph animation* [Junker, 2006].

- *Morph animation*

Morph animation é o método de calcular a posição dos vértices mais simples e mais directo dos dois tipos de *vertex animations*. Em vez de calcular a posição dos vértices em função da posição do esqueleto, a posição actual dos vértices é armazenada como uma sequência de “alterações alvo” numa animação. Neste processo, as posições dos vértices de uma malha são ajustados e a *keyframe* é armazenada, contendo a posição absoluta de cada um dos vértices nas malhas animadas. Como se pode imaginar, isto requer o uso intensivo de recursos, pois cada frame chave contém uma cópia completa da informação dos vértices. Contudo, isto é um processo de animação computacionalmente eficiente, pois é uma questão de calcular a interpolação das posições intermédias para cada vértice de uma malha, de uma frame para a seguinte.

O principal inconveniente das animações do tipo *morph animation* é que estas não podem ser misturadas entre si, uma vez que a posição dos vértices é armazenada como absoluta, o que se opõe à mistura de informação diferente referente ao mesmo vértice (por exemplo, duas sequências que animam a mesma mão) [Junker, 2006].

- **Pose animation**

As animações do tipo *pose animation* baseiam-se no facto das várias poses da animação serem armazenadas em sequências de animação. Estas poses podem ser misturadas de forma a criar animações mais complexas e realistas do que as conseguidas sem recorrer a mistura. A maior diferença é que a posição de cada vértice da malha, em vez de ser guardada como posição absoluta, é guardada como o deslocamento a partir das restantes poses. Além disso, apenas são guardados os dados dos vértices que sofreram alterações (ao contrário das *morph animations* em que a informação de todos os vértices da malha é armazenada). Como resultado, as animações do tipo *pose animation* não só são mais flexíveis do que as do tipo *morph animation*, como também são mais eficazes a nível de recursos [Junker, 2006].

3.5 CEGUI

No desenvolvimento de aplicações gráficas, pode ser necessária a configuração de um interface gráfico do utilizador (GUI) que consiste num tipo de interface que permite que os utilizadores interajam com as aplicações. Uma GUI disponibiliza ícones gráficos, e indicadores gráficos, ao contrário das interfaces baseadas em texto, onde é necessário introduzir os comandos para representar a informação e as acções disponíveis para o utilizador. Numa GUI as acções são normalmente realizadas através da manipulação directa dos elementos gráficos.

A CEGUI foi criada por Paul Turner (“Crazy Eddie”), estando actualmente disponível a versão “0.7.4”. A Crazy Eddie’s GUI System (CEGUI) é uma poderosa GUI livre, que disponibiliza *widgets* (janelas, botões, caixas de texto, etc.) para aplicações gráficas onde tais funcionalidades não estão nativamente disponíveis, ou são gravemente carentes. Também é possível personalizar a interface CEGUI recorrendo a *skins*, através dos quais pode ser alterada a sua aparência de forma a uniformizá-la visualmente com os outros elementos da aplicação. A biblioteca é orientada a objectos e desenvolvida em C++, sendo destinada a programadores de jogos que pretendem orientar os seus esforços para o desenvolvimento de bons jogos, e não para a criação de subsistemas da interface gráfica [CEGUI].

A CEGUI é um sistema multi-plataforma, podendo ser utilizado com Linux, Windows e Mac OS X. É compatível com o motor gráfico *Ogre 3D*, e com o motor gráfico *Irrlicht*. Também suporta nativamente o Microsoft® DirectX®, e o OpenGL.

Como o *Ogre 3D* oferece apenas o ambiente gráfico, sem nenhum tipo de suporte GUI, existe aqui uma lacuna. De modo a evitar a construção de uma interface de raiz, esta lacuna pode ser preenchida pela CEGUI, de forma a focar os esforços no desenvolvimento da aplicação gráfica principal. De forma simplificada, a CEGUI funciona essencialmente acrescentando uma segunda cena à janela de exibição que é renderizada depois pelo *Ogre 3D*. Esta segunda janela é baseada num ficheiro XML, bastando apenas especificar qual o arquivo a ser carregado para que a interface seja exibida.

Embora a CEGUI seja a GUI com mais documentação disponível relativamente à integração com o *Ogre 3D*, neste momento estão em desenvolvimento várias interfaces gráficas compatíveis com esta API, entre as quais se pode destacar a *Quick GUI*, que está desenhada para ser fácil de usar e, ao mesmo tempo, eficiente e poderosa. Esta GUI começa a ser bastante referenciada na comunidade de desenvolvimento em *Ogre 3D*, estando já disponível alguma documentação de apoio à sua utilização [QuickGUI].

3.6 SUMÁRIO

A evolução do processo de animação foi sempre acompanhada pela evolução das ferramentas vocacionadas para a criação destas animações. Estas ferramentas são constantemente adaptadas ao aparecimento de novas técnicas de animação. Apesar de actualmente o termo Realidade Virtual não ser muito utilizado, a quantidade e variedade de ferramentas de síntese de ambientes virtuais tem crescido largamente nos últimos anos. No presente capítulo foram expostas as diversas fases do processo de animação computacional adoptado, onde estão descritas as operações realizadas e as tecnologias utilizadas para as realizar.

4 SOLUÇÃO PROPOSTA

Nos capítulos anteriores foram introduzidos os conceitos e as tecnologias estudadas. Foi também apresentado o projecto *VirtualActor*, no qual foi integrada a investigação e o desenvolvimento obtidos no âmbito desta dissertação. No presente capítulo são descritas as diferentes etapas que resultaram na alternativa proposta de implementação de um sistema de animação de personagens, onde se pretende dotar personagens virtuais com a capacidade de representar comportamentos e emoções específicos de um actor humano.

4.1 CAPTURA ÓPTICA DE MOVIMENTOS

A captura de movimentos utilizando um sistema óptico é um método que permite uma grande precisão na captura de certos movimentos, quando se usa o sistema correctamente. Não é possível obter o *feedback* imediato do movimento no personagem de destino, mas o auxílio visual é possível em tempo quase real num personagem mais simples. Contudo, os dados adquiridos opticamente exigem um extenso processamento posterior para que se tornem utilizáveis.

Nesta dissertação foi utilizado o sistema óptico de *Motion Capture* instalado na sede do Centro de Computação Gráfica, no Campus de Azurém da Universidade do Minho, em Guimarães. A informação é gravada utilizando um Sistema *Vicon MX* composto por 6 câmaras capazes de captar 250 fps na máxima resolução, podendo atingir até 2000 fps. O actor veste uma roupa especial onde são

colocados quarenta e um marcadores reflectores, conforme o esqueleto hierárquico da ferramenta de edição e processamento *Vicon iQ*. Um sistema típico de captura óptica de movimento é baseado num único computador que controla a entrada varias câmaras. Uma câmara como a *Vicon MX* contém uma matriz de pixéis cuja resolução varia, normalmente, entre a resolução 128 x 128 e 4096 x 4096.

A taxa de amostragem por segundo, ou *frame rate*, tem de ser suficiente rápida para captar as nuances de movimentos muito rápidos. Na maioria dos casos, 512 amostras por segundo é mais do que suficiente. Pelos padrões actuais, uma câmara com uma resolução de 4096 x 4096 seria capaz de produzir menos de uma frame por segundo, mas isso está a mudar muito rapidamente [Parent et al., 2009].

O número de câmaras normalmente utilizado é, no mínimo, quatro e no máximo trinta e duas, e estas capturam a posição de marcadores em qualquer lugar com velocidades entre 30 e 1000 amostras por segundo. As câmaras são normalmente enquadradas com as suas próprias fontes de luz, que criam uma reflexão direccionada a partir dos marcadores. Estes são geralmente esferas cobertas com um material reflector. São preferidas fontes de luz infravermelha, pois estas criam menos distorção visual para o actor. As esferas dos marcadores podem variar de poucos milímetros de diâmetro, para pequenas áreas de captura, até alguns centímetros. O sistema *Vicon MX*, ilustrado na Figura 49, é um exemplo de um sistema óptico, que pode acomodar até 24 câmaras.



Figura 49 - Sistema Vicon MX

O sistema óptico deve ser calibrado, tendo todas as câmaras registado o movimento de um objecto com dimensões conhecidas que o software é capaz de reconhecer, como uma varinha com marcadores. Ao combinar os pontos de vista de todas as câmaras com o objecto de dimensões conhecidas, pode ser determinada a posição exacta de cada câmara no espaço [Parent et al., 2009]. A Figura 50 ilustra duas

varinhas de calibração do sistema, uma de 240mm (A), e outra de 120mm (B). O processo de calibração encontra-se detalhado no “Anexo A”.

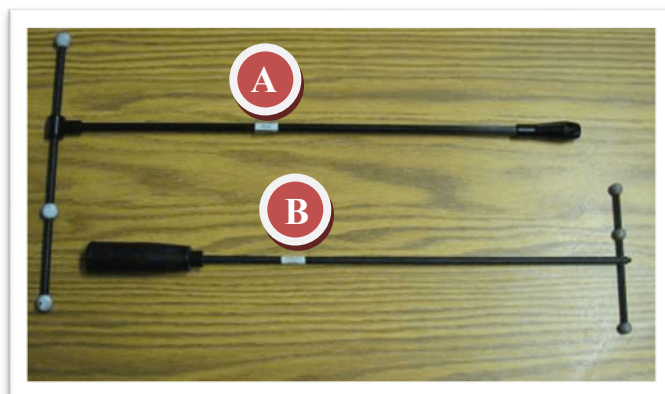


Figura 50 – (A) varinha de 240 mm; (B) varinha de calibração de 120 mm

Se alguma das câmaras for afastada da posição onde foi calibrada, nem que esse afastamento tenha sido mínimo, deve ser realizada uma nova calibração. Se o estúdio não possuir estruturas físicas de suporte das câmaras, de modo que estas não alterem a sua posição, devido a qualquer tipo de movimento ou vibração, será melhor recalibrar o sistema após alguns minutos de captura.

Para monitorizar a posição de um único ponto 3D são necessários pelo menos dois pontos de vista, de modo que, as restantes câmaras são essenciais para garantir que a cada instante pelo menos duas delas monitorizam a posição de cada um dos marcadores. Isso não significa que quanto mais câmaras melhor, pois cada câmara adicional aumenta o tempo de processamento. Existem outros métodos para minimizar oclusões que são implementados em software e que são utilizados durante o pós-processamento. A solução ideal é diferente em cada caso, dependendo, do tipo, da velocidade, e da duração de cada movimento, bem como da luz disponível.

Uma vez que os pontos de vista da câmara estejam digitalizados no computador, é iniciado o pós-processamento. O primeiro passo consiste na criação de uma reprodução limpa, que contenha apenas informação relativa aos marcadores. São utilizados diferentes métodos de processamento de imagem para minimizar o ruído e isolar os marcadores, separando-os da restante informação. A abordagem mais básica consiste em separar todos os grupos de pixéis que ultrapassam um determinado limiar de luminosidade. É natural que algumas frames contenham informação incompleta, para completar a informação de uma determinada frame a aplicação *Vicon iQ* possui a capacidade de utilizar informação de frames adjacentes.

O segundo passo consiste na determinação das coordenadas 2D de cada marcador conforme o ponto de vista de cada câmara. Para obter as coordenadas 3D de cada marcador, as coordenadas 2D são combinadas com as coordenadas da própria câmara e com os pontos de vista das restantes câmaras. O terceiro passo visa identificar realmente cada marcador durante a sequência. Esta etapa requer mais

assistência do operador, uma vez que a atribuição inicial (rotulagem) de cada marcador tem de ser registada manualmente. A Figura 51 ilustra a distribuição de marcadores adoptada.

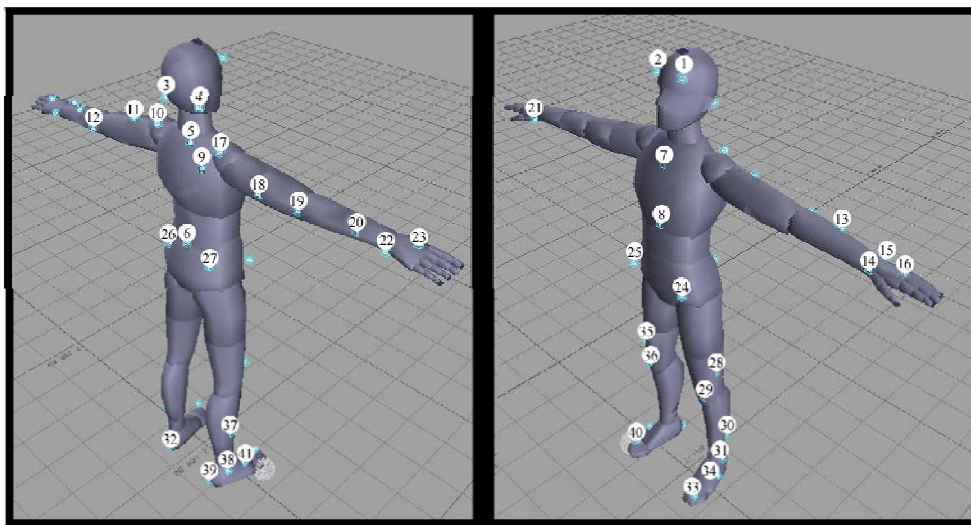


Figura 51 - Distribuição de Marcadores

Cada um dos marcadores utilizados é reconhecido na aplicação *Vicon iQ* pelo seu rótulo. No “Anexo A” pode obter-se informação mais detalhada relativamente à correspondência entre os rótulos e a respectiva localização no corpo do actor.

Depois desta atribuição, o software tenta determinar a sequência restante, embora por vezes deixe de conseguir monitorizar alguns marcadores devido à oclusão, ponto em que o operador deve voltar a atribuir os marcadores em questão, para que o cálculo possa prosseguir. Este processo repete-se até que a sequência tenha todos os marcadores atribuídos. No final é criado um ficheiro que contém os dados posicionais de todos os marcadores. O ficheiro produzido por este processo contém uma sequência de posições globais dos marcadores ao longo do tempo, o que significa que em cada frame apenas são listadas as coordenadas cartesianas (x, y, z) de cada marcador e não é incluída qualquer hierarquia ou rotação de membros. Os realizadores técnicos experientes podem beneficiar da utilização dos dados deste ficheiro, pois este permite controlar melhor o que pode ser feito na configuração do personagem. No entanto, para o utilizador médio os dados devem ser tratados, pelo menos até à inclusão de um esqueleto hierárquico, através do qual são definidas as rotações dos membros [Parent et al., 2009].

É muito importante ter em conta que o posicionamento dos marcadores é relativo, e não absoluto. Ou seja, não é necessário seguir exactamente um diagrama, desde que cada marcador esteja posicionado sobre o mesmo osso que no diagrama. Deve evitar-se colocar marcadores sobre os músculos, pois estes vão fazer com que os marcadores se movam independentemente dos ossos. Os marcadores

devem ser colocados nas articulações. Uma forma de conseguir um bom posicionamento do marcador é rodar a articulação do actor até encontrar um ponto que gire menos, de preferência sobre um osso. É exactamente nesse local onde deve ser colocado o marcador. Os marcadores são geralmente colocados do lado de fora das articulações para que sejam mais visíveis para as câmaras. Também é sugerida a colocação dos marcadores de forma assimétrica, de modo a auxiliar a aplicação *Vicon iQ* na distinção do lado esquerdo e direito do personagem.

4.2 COMPOSIÇÃO DE ANIMAÇÕES DE POSE

Na demonstração da solução proposta, são alcançados vários fonemas e diversas expressões faciais através da mistura de múltiplas poses. A Figura 52 ilustra a mistura de várias poses da face do personagem virtual, de forma a recriar uma expressão facial inteiramente nova, mais complexa e porventura mais interessante.

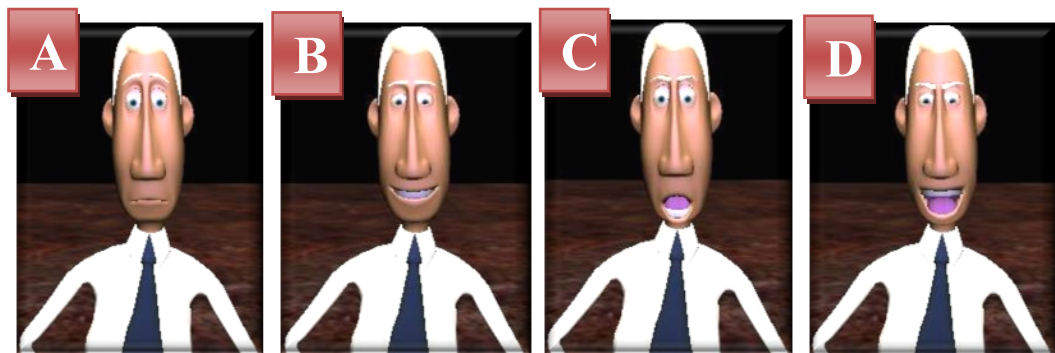


Figura 52 – Expressões faciais: (a) pose *neutra*; (b) pose *feliz*; (c) pose *surpreendido*; (d) mistura pose *feliz* e *surpreendido*

Neste caso, o resultado da Figura 52 (D) é uma expressão facial composta pela mistura de outras duas (*feliz* e *surpreendido*). É importante referir que elementos da face tais como, as sobrancelhas, o sobre olho, os lábios, os dentes e a língua obtêm uma nova configuração, influenciada pelas poses *feliz* e *surpreendido*. Por outro lado a configuração dos olhos mantém-se pois a sua orientação é a mesma nas poses *neutra*, *feliz* e *surpreendido*.

As frames chave das *pose animation* referem uma ou, mais trajectórias de pose, e cada referência tem um peso que influencia a pose resultante. Voltando a Figura 52 (D), esta é obtida com 60% de influência da pose *feliz* e 40% da pose *surpreendido*. Quando as poses são misturadas, a influência dos pesos determina a forma como o deslocamento dos vértices afecta a sua posição final.

A Figura 53 ilustra um conjunto de fonemas simplificados da língua inglesa, que têm por base animações de pose. Estes podem ser utilizados como objectivos de transformação. Ao combinar vários

fonemas é possível construir palavras. Ou seja, a combinação de fonemas é extremamente útil pois permite geração dinâmica de diálogos entre personagens.

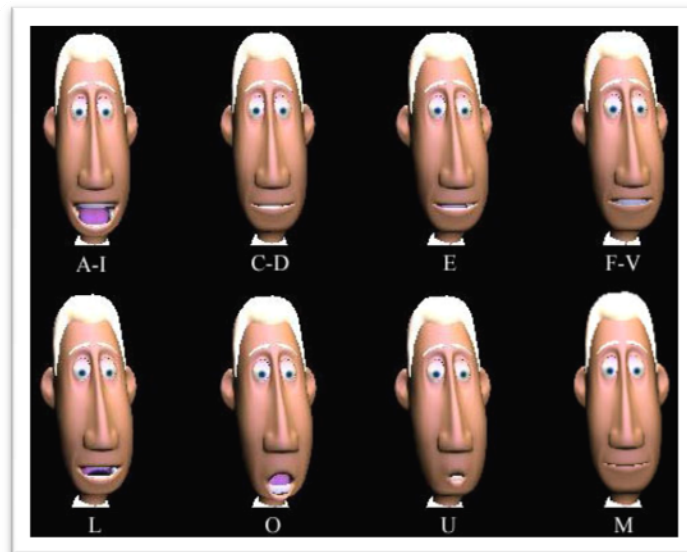


Figura 53 - Construção de fonemas da língua inglesa, para a animação de diálogos

4.3 MISTURA DE DIFERENTES TIPOS DE ANIMAÇÕES

Ao identificar diferentes movimentos para o personagem executar, pretende-se que seja possível fazer uma combinação de movimentos através da mistura de dois movimentos, ou fazer uma transição de um movimento para outro. O *Ogre 3D* possibilita a mistura de diversos tipos de animações de forma a permitir a criação de uma cena com animações complexas. A exceção é que as animações do tipo *morph* não podem ser misturadas com outras animações do tipo *morph* nem com animações do tipo *pose*.

4.3.1 Composição de animações de pose e de esqueleto

Quando se misturam animações de vértices e de esqueleto, o *Ogre 3D* calculará primeiro a mistura das animações de vértices, só depois aplica as deformações provocadas pelo esqueleto a esses vértices. Como se verifica na Figura 54, a aplicação conjunta de misturas de poses e de animação de esqueletos, permite a conjugação de animações de corpo inteiro com expressões faciais complexas, por forma a obter a animação realista de um personagem. Ou seja, esta solução permite animar a face do personagem utilizando animação de pose, enquanto o restante corpo utiliza a animação mais comum e eficiente que é a animação de esqueleto.

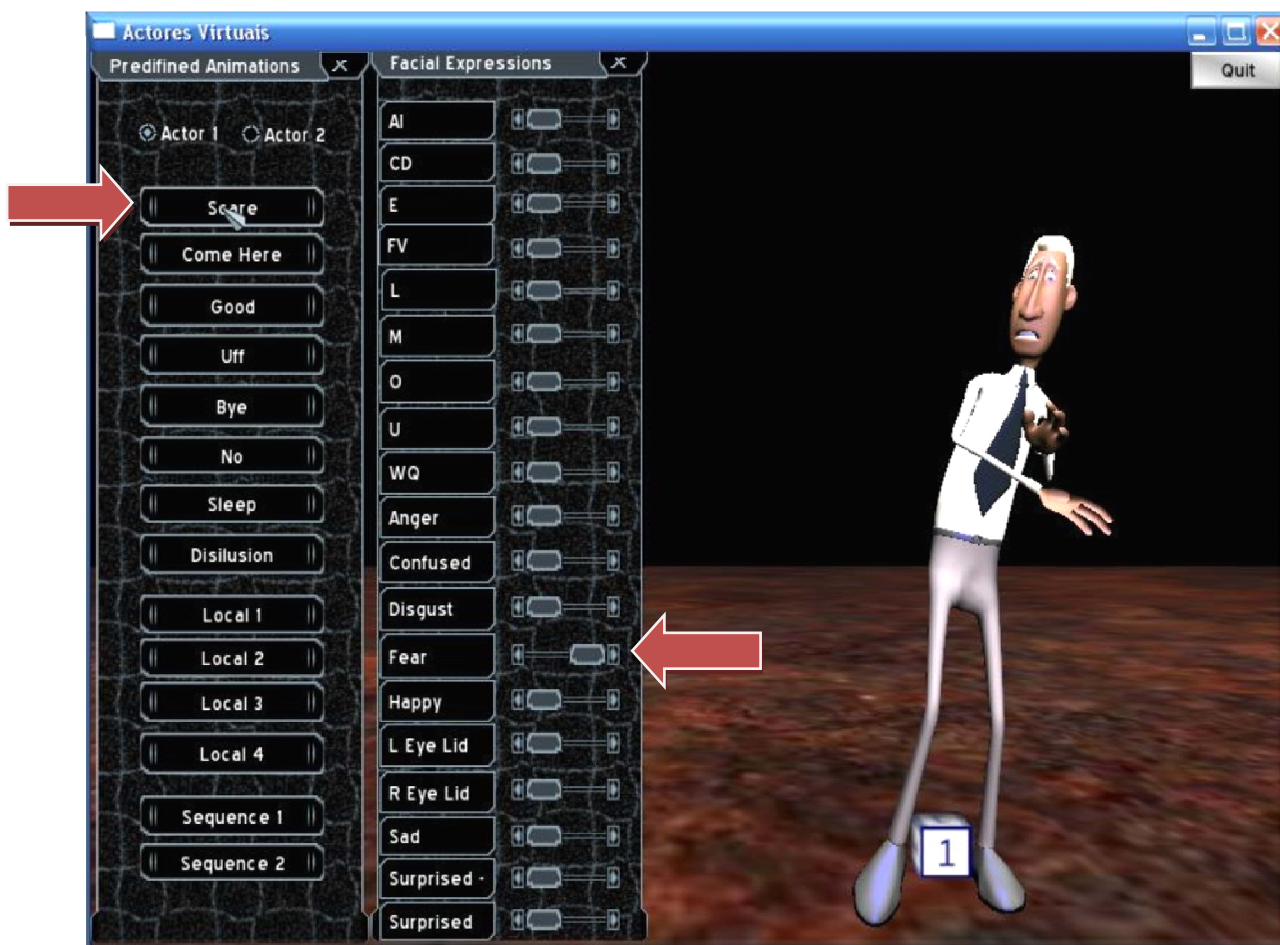


Figura 54 - Composição de *Skeletal Animation* com *Pose Animation*

A composição de animações de pose e de esqueleto pode também ser útil para aproximar o movimento da roupa do personagem. Neste caso, como foi escolhido um personagem de estilo *cartoon* não é importante a simulação da roupa, mas caso fosse utilizado um personagem de estilo mais realista, seria necessário adoptar alguma estratégia para simular o comportamento da roupa. Uma boa escolha seria a utilização de animações de pose para a animar a “deformação” da roupa, podendo então efectuar-se a mistura desta “deformação” com o movimento do esqueleto do personagem, evitando assim os elevados custos de processamento de um simulador de física em tempo real.

4.3.2 Composição de múltiplas animações de esqueleto

Ao misturar várias animações de esqueleto, é possível executar vários movimentos em simultâneo. A mistura de animações, em tempo real, pode ser obtida através da mistura baseada no peso de cada animação. Este processo consiste em activar sobre um personagem duas animações simultaneamente, o que significa que enquanto ambas as animações estiverem a ser combinadas, o movimento resultante será uma mistura destas, podendo sofrer uma influência diferente de cada uma delas. Cada frame do

movimento resultante é obtida através da mistura da orientação de cada articulação do esqueleto do personagem, com a orientação das articulações correspondentes no outro movimento. Por exemplo, a orientação da articulação que representa o pescoço na primeira animação é misturada com a orientação da articulação que representa o pescoço na segunda animação, sendo este processo repetido para cada uma das restantes articulações.

Existem duas regras de ouro que devem ser respeitadas, de forma a misturar com sucesso dois ciclos de animação: os ciclos devem ter o mesmo tamanho; e as próprias acções não devem ser muito diferentes. Na Figura 55 está ilustrado o processo de mistura de animações e a utilização de *Timewarping* para alinhar as frames semelhantes dos movimentos interpolados, garantindo animações do mesmo tamanho, de forma a reforçar a correspondência temporal e a reduzir o aparecimento de artefactos não naturais. As setas vermelhas e brancas representam respectivamente o peso das animações um e dois na composição da animação resultante.

Dado que no âmbito desta dissertação, a composição de animações é utilizada apenas como método para a construção de transições suaves entre movimentos, a primeira regra de ouro referida anteriormente não se aplica, pois ambas as animações só contribuem simultaneamente para a animação resultante durante a fase de transição. Ou seja, após a fase de transição apenas uma das animações activas influenciará o movimento do personagem; logo, não foi necessária a implementação de um algoritmo de *Timewarping*.

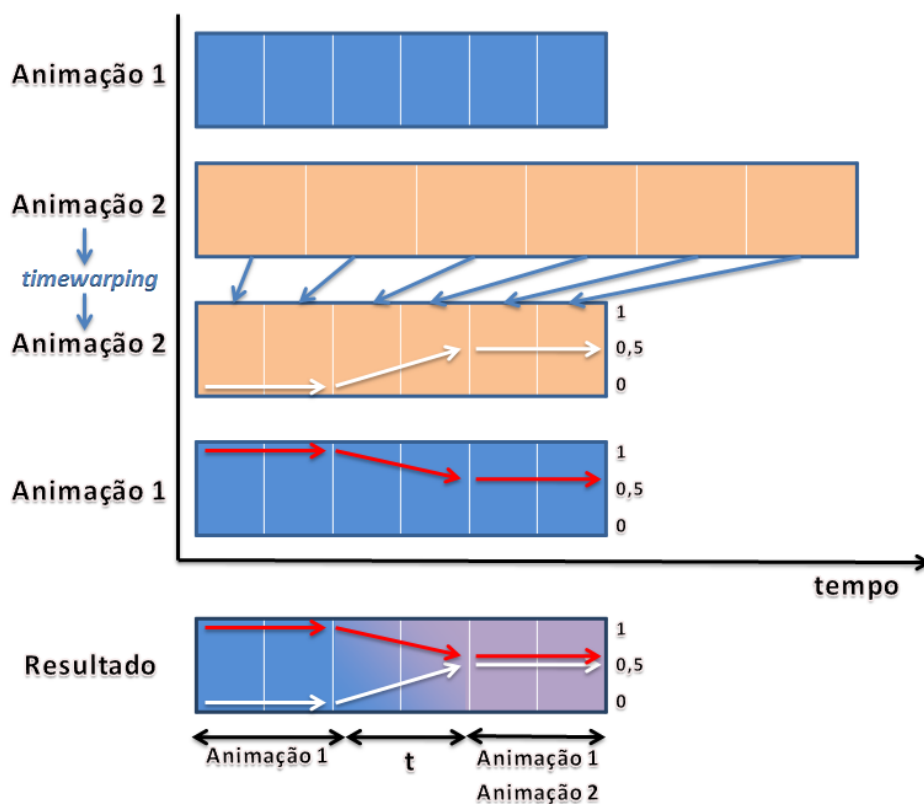


Figura 55 - Composição de animações

4.3.2.1 Transição de movimentos

A transição entre movimentos baseia-se na mistura de animações. A diferença reside no facto de a transição ser uma mistura temporária. A transição é normalmente utilizada para comutar entre um movimento e o movimento seguinte. Quando se toma a decisão de alternar de um movimento para outro, é iniciada uma mistura de movimentos, mas apenas durante o período de transição determinado. Este período poderá ser diferente, dependendo dos movimentos a serem misturados.

Para gerar a transição entre os dois movimentos, são criadas novas frames através de interpolação. Em cada uma destas novas frames é aplicada uma transformação a cada osso do esqueleto, de modo a que cada um obtenha uma nova posição e orientação. Assim, é necessário interpolar a posição e a orientação de cada articulação. Para a interpolação da posição é utilizada interpolação linear. Para a interpolação da rotação das articulações, a utilização de quatérnions revela-se a melhor opção, sendo criado um quatérnio apropriado para cada articulação do personagem.

A Figura 56 ilustra a transição entre duas animações de movimentos através da interpolação da orientação de cada uma das articulações que compõem o esqueleto. As setas brancas e vermelhas representam respectivamente o peso das animações um e dois na composição da animação resultante. Esses pesos são determinados a partir do período de transição; portanto, quando a transição é iniciada, a animação que está a terminar tem o peso a um, e a animação que está a iniciar tem o peso a zero. Depois de decorrido o período de transição, a animação que terminou tem o peso a zero e a animação que iniciou tem o peso a um. Isso significou o alternar de um movimento para outro.

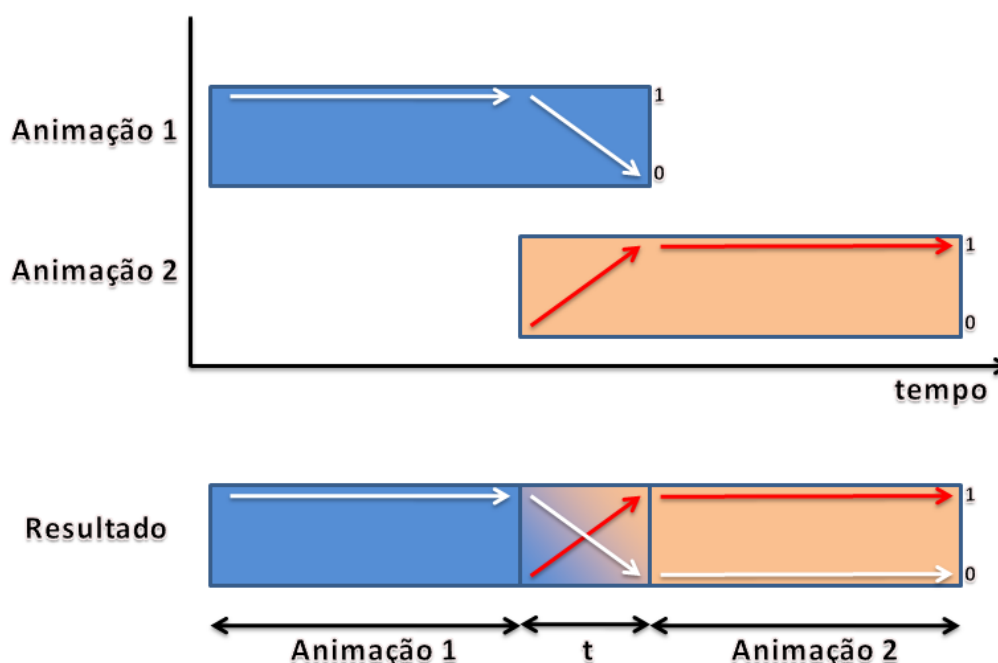


Figura 56 - Transição entre duas animações através da interpolação da orientação de cada articulação

Durante o período de transição, um determinado osso terá a posição e a orientação, mais aproximada da animação com maior peso. Na Figura 57 está ilustrado um caso concreto de transição entre animações. Na primeira “frame”, o personagem permanece em modo *descanso* até ocorrer a primeira fase de transição para o modo de *negação* (zona avermelhada). Durante a execução da animação de *negação* a dado momento o personagem altera o seu estado para o modo *despedida*, cuja fase de transição está representada a cor amarelada.



Figura 57 - Mistura de animações de esqueleto

Graças à utilização de um sistema de transição de animações recorrendo à mistura de animações, é evitado o efeito indesejável, que resulta de transitar instantaneamente de uma animação para a seguinte. Neste caso, podem ser misturadas ambas as animações, sendo aumentado gradualmente o peso da animação de *despedida* e simultaneamente diminuído o peso da animação de *negação*. O resultado final é uma transição harmoniosa.

4.4 MÓDULO DE LOCOMOÇÃO

Para a simulação do comportamento de personagens interactivos realistas, a navegação, a detecção de obstáculos e a capacidade de os evitar, devem ser incluídas num sistema de animação em tempo real.

O adicionar de navegação dinâmica torna possível simular avatares que se movem para destinos específicos no espaço, ou para evitar obstáculos, ou outros avatares. Na aplicação desenvolvida o controlo da locomoção dispõe apenas das funcionalidades básicas alcançadas através da parametrização da velocidade e do ângulo de viragem.

Ao conectar animações sucessivamente deve também ter-se em conta o tempo, pois estas animações normalmente têm diferentes velocidades. Esta diferença tem de ser manipulada para poderem ser geradas transições de movimentos convincentes. Um exemplo de transição onde a velocidade de cada animação não é a mesma é a transição de *caminhada* para *corrida*. Neste caso, foi necessário criar uma solução para minimizar o efeito de arrastamento de pés abordado na secção 2.9 e conhecido como *footskate*.

A solução encontrada passou por dividir o movimento de *caminhada* e de *corrida* em várias secções independentes, cada uma delas com um deslocamento do personagem ajustado à velocidade. Na Figura 58 é possível visualizar o ajuste da velocidade (representada pela seta branca) ao mesmo tempo que ocorrem as transições entre movimentos. Neste caso, ao iniciar o movimento de *corrida* existe uma primeira fase em que o personagem inicia o deslocamento com a velocidade zero. Esta fase termina assim que o personagem atinge a velocidade de *corrida*, na qual permanece durante a segunda fase. A terceira fase do movimento acontece quando o personagem já se encontra relativamente próximo do local de destino e inicia a desaceleração para chegar ao destino com a velocidade zero.

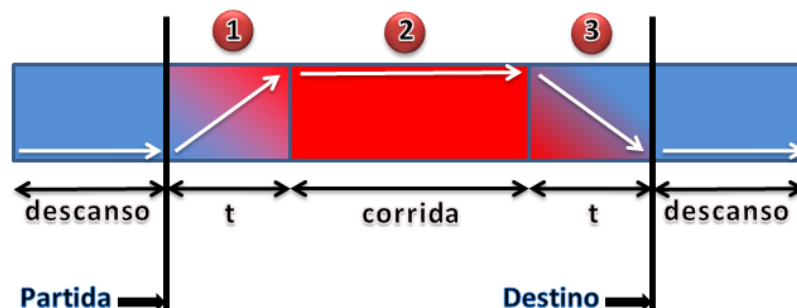


Figura 58 - Ajuste da velocidade de corrida

A Figura 59 ilustra uma situação em que o movimento do personagem é condicionado por diversas variações de velocidade. Neste caso, ao iniciar o movimento de *caminhada* existe uma primeira fase em que o personagem inicia o deslocamento com a velocidade zero, esta fase termina assim que este atinge a velocidade de *caminhada*, na qual permanece durante a segunda fase. Na terceira etapa a velocidade volta a aumentar progressivamente, ao mesmo tempo que é aumentado o peso da animação de *corrida*, esta etapa termina ao ser atingida a velocidade de *corrida*, na qual permanece durante a quarta fase. A última fase do movimento acontece quando o personagem já se encontra relativamente próximo do local de destino e inicia a desaceleração para chegar ao destino com a velocidade zero.

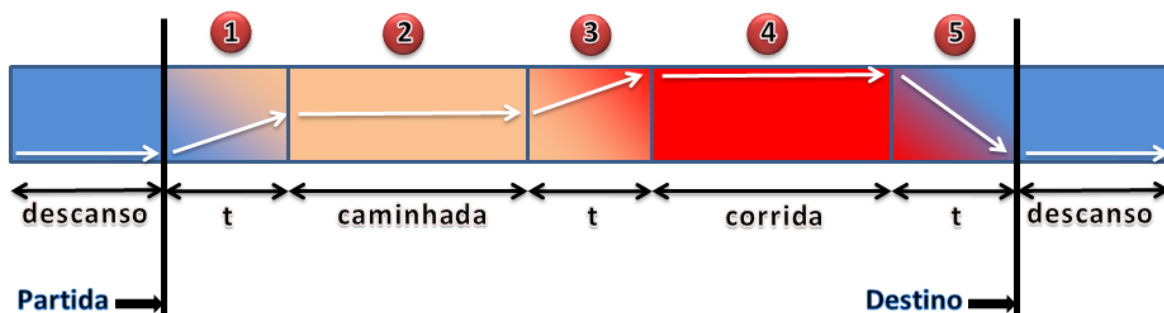


Figura 59 - Ajuste das velocidades de caminhada e corrida

4.4.1 Interpolação de Rotações 3D

Ao permitir a interpolação entre o movimento em linha recta e em curva, o sistema é capaz de simular a locomoção com diferentes curvaturas. A interpolação entre duas orientações recorrendo a quaterniões revelou-se a forma mais suave de interpolar ângulos. Os quaterniões são utilizados para acompanhar a forma como o personagem é posicionado no espaço e para o orientar na direcção pretendida, através do método conhecido como *slerp* (ver secção 2.10.3). Ou seja, são utilizados quaterniões para representar a rotação a partir da direcção em que o personagem está orientado, para a direcção para a qual se pretende que este seja orientado. Na Figura 60 é possível verificar a interpolação de rotações 3D no movimento do personagem virtual num caso concreto, em que partiu da localização dois com destino à localização quatro, sensivelmente a meio deste percurso, o utilizador define como ponto de destino a localização três. Assim verifica-se que, recorrendo a interpolação linear esférica, é possível ajustar a trajectória do personagem, orientando-o de forma a descrever um arco em direcção ao destino.




Figura 60 - Interpolação de rotações 3D utilizando SLERP



4.5 MÁQUINA DE ESTADOS

Os personagens animados num jogo comportam-se geralmente segundo um conjunto finito de estados. Estes exercem as mesmas actividades até que algum evento ou influência desencadeie a transição entre estados.

A mistura de animações é bastante útil para combinar alguns movimentos, mas não resolve o problema do sequenciamento de diversas animações ao longo do tempo. Para isso, é necessário utilizar uma máquina de estados finita (ver secção 2.11). Como primeira tentativa para sequenciar várias animações, foi implementado um protótipo de máquina de estados finita que permite a interacção do personagem com o utilizador e com o ambiente. Esta máquina de estados permite sequenciar as animações ao longo do tempo, sendo definida como um grafo conectado de estados e transições. O grafo, construído manualmente, é do tipo *move tree* e tem como função organizar a estrutura de animações. Esta construção poderia teoricamente ser automatizada (*motion graph*), mas o foco neste caso foi o planeamento eficiente das restrições dos movimentos, em vez da construção automática do grafo. No contexto do sequenciamento de animações, os estados constituem a representação individual de cliques de animação e as transições são desencadeadas por algum tipo de evento.

A Figura 61 ilustra o diagrama da máquina de estados finita, onde os nomes dos estados indicam claramente o movimento que se pretende representar. A máquina é composta por 11 estados: *descanso*; *caminhada*; *corrida*; *felicitação*; *chamar*; *susto*; *alivio*; *negação*; *despedida*; *sono* e *desilusão*. Para todos os eventos é permitido definir uma sequência de animações, bem como a frame onde se inicia a alteração determinística da transição entre estados de animação.

A máquina de estados é composta por dois grupos de animações. O primeiro grupo de animações, assinalado com  na Figura 61, (*descanso*; *caminhada*; *corrida*) contém as animações base, que funcionam em ciclo até que ocorra algum evento que obrigue a mudança de estado. Por exemplo se o personagem permanecer no estado *caminhada* este transitará automaticamente para o estado de *descanso* assim que atinja o seu destino. A partir do grupo de animações base é permitida a transição para todas as outras animações.

As animações do segundo grupo, assinalado com  na Figura 61, (*despedida*; *felicitação*; *negação*; *chamar*; *susto*; *alivio*; *desilusão*; *sono*) não funcionam em ciclo, de modo que assim que é atingido o seu final ocorre a transição automática para a animação *descanso* (primeiro grupo). Dentro do segundo grupo existe um subgrupo de animações, assinalado com  na Figura 61, (*desilusão*; *sono*) cujas animações são as únicas que podem ser acedidas directamente pelas seguintes animações do segundo grupo: *despedida*; *negação*. Esta transição é permitida devido à semelhança entre as acções a executar.

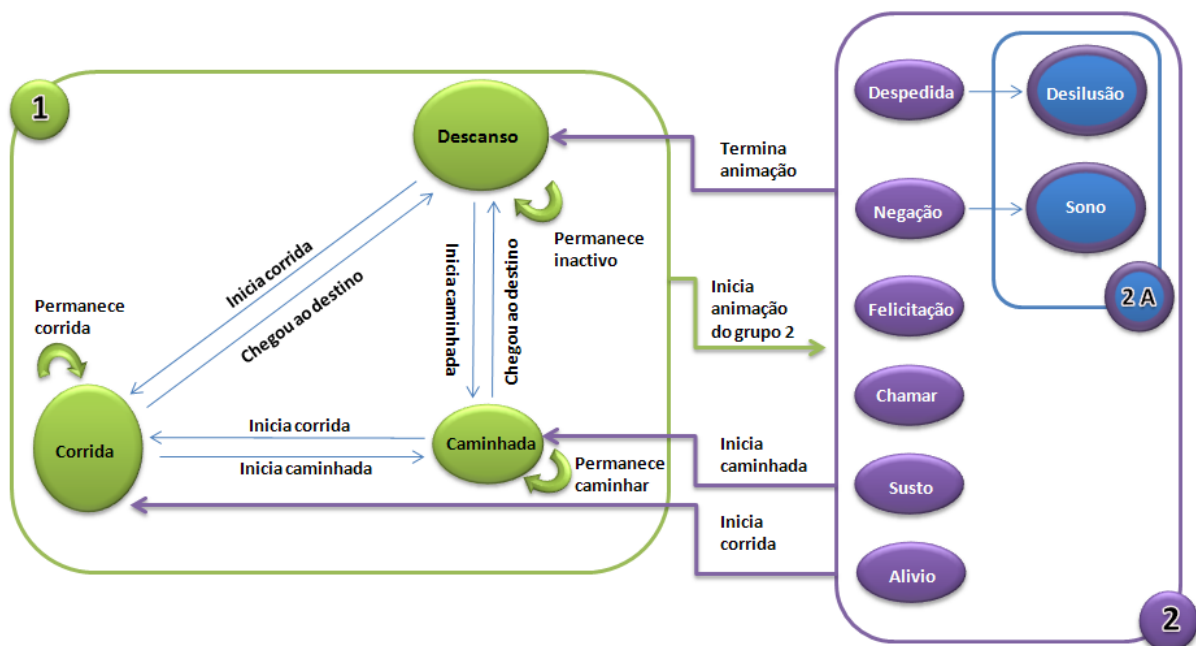


Figura 61 - Diagrama de estados

4.6 SUMÁRIO

No presente capítulo foram apresentadas as várias etapas do processo de captura de movimentos levado a cabo para a obtenção de informação de movimento. Estes dados são tratados utilizando as ferramentas e métodos descritas no capítulo anterior, de forma a construir uma biblioteca de desempenho referente a um personagem virtual. Foram ainda descritas as diferentes etapas que resultaram na alternativa proposta de implementação de um sistema de animação de personagens, onde se pretende dotar personagens virtuais com a capacidade de representar comportamentos e emoções específicos de um actor humano. Estas etapas incluem a criação de um método de mistura de animações que é bastante útil para efectuar a transição entre movimentos e a implementação de funcionalidades básicas de controlo da locomoção alcançadas através da parametrização da velocidade e do ângulo de viragem. E a implementação de uma FSM para resolver o problema de sequenciamento de diversas animações ao longo do tempo.

5 ANÁLISE DE RESULTADOS

No presente capítulo é apresentado o processo de avaliação do sistema proposto: desde a integração do sistema de animação no software de criação de filmes de animação de actores virtuais *VirtualActor* (ver secção 1.1.1), à criação das simulações, culminando na exposição e análise dos resultados obtidos. O *VirtualActor* foi utilizado para realizar simulações de validação do sistema proposto. Para tal foi construído um “cenário” básico e diversas “histórias”, de modo a testar várias situações específicas da animação de personagens virtuais.

Os testes efectuados visaram a determinação da qualidade das transições entre animações. Para quantificar a qualidade das transições foi criada a Tabela 1, cuja escala varia entre um e dez. O valor um corresponde a uma transição com graves deficiências e o valor dez corresponde a uma transição muito suave.

Tabela 1 - Classificação qualitativa de transições entre animações

1	2	3	4	5
Transição com graves deficiências	Transição com problemas de deslizamento em ambos os pés		Transição com problemas de deslizamento de um dos pés	
6	7	8	9	10
Não se verifica deslizamento de pés. Mas verifica-se a ocorrência de poses intermedias irrealistas		Transição suave		Transição muito suave

Na Figura 62 é possível visualizar o cenário criado. Este consiste num caminho que o personagem percorre e ao longo do qual executa diversos movimentos. Como o software *VirtualActor* dispõe de uma linha de tempo, é possível construir uma “história” que, no caso dos testes efectuados, consiste na criação de uma sequência lógica de cliques de animação.

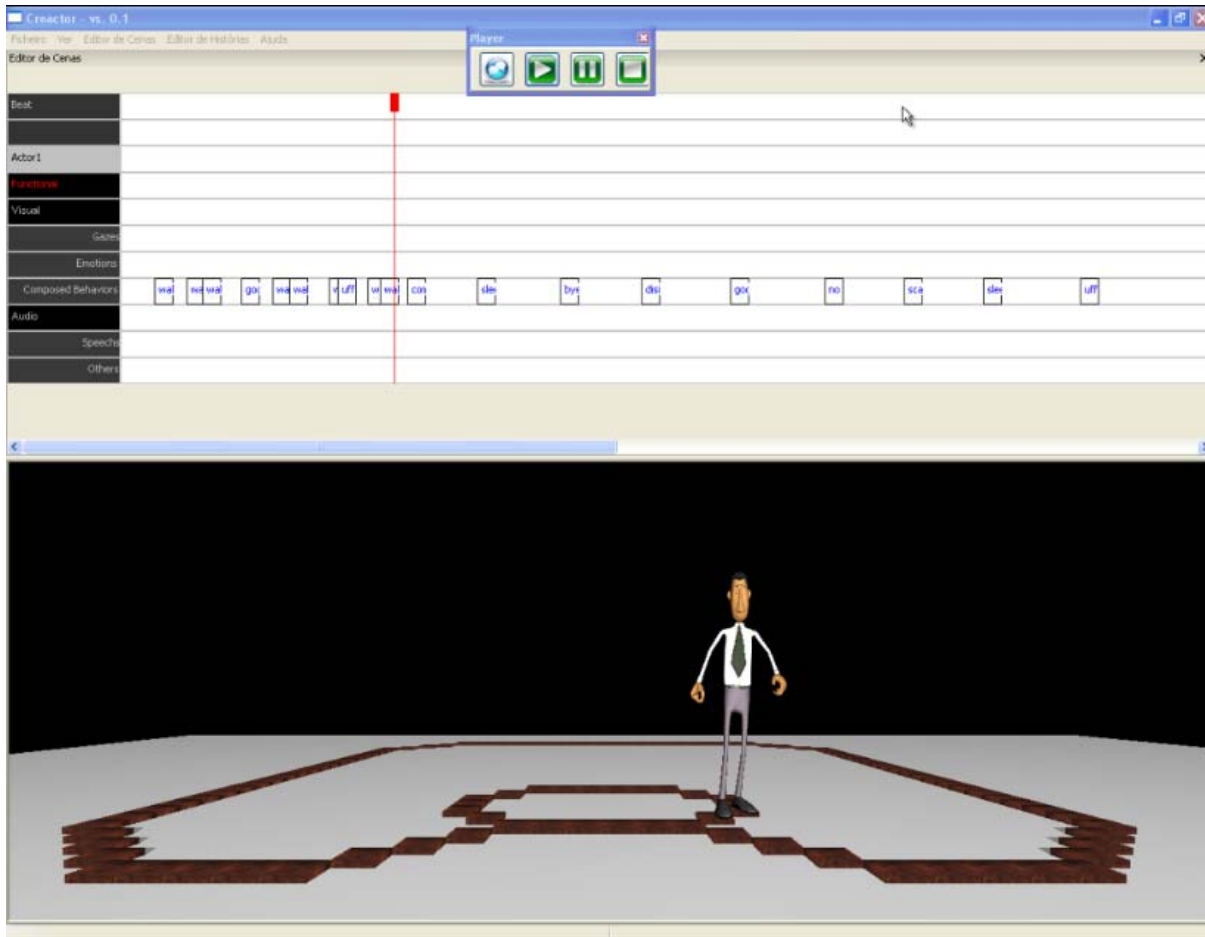


Figura 62 - Teste do sistema proposto integrado na aplicação *VirtualActor*

A Tabela 2 ilustra os resultados obtidos nos testes efectuados. Esta tabela indica a qualidade média de cada uma das transições, devendo a leitura desta ser feita tendo em conta que partindo de cada uma das animações origem será possível transitar para várias animações destino. A qualidade de cada uma das animações é representada pela classificação e pela respectiva cor (conforme a Tabela 1). As células pretas indicam que a transição não é permitida pela máquina de estados (ver diagrama da Figura 61).

Tabela 2 - Resultados obtidos nas simulações (transições entre animações)

		DESTINO										
		bye	comeHere	disillusion	good	idle	no	run	scare	sleep	uff	walk
ORIGEM	bye			6		6		6				9
	comeHere					9		9				8
	disillusion					8		9				8,5
	good					6		6				8
	idle	9	8	10	7	10	8	10	9	9	9	10
	no					9		10		7		8
	run	5	2	2	5	6	3	8	3	5	3	8
	scare					9		7				7
	sleep					9		9				8
	uff					9		9				8
	walk	5	5	5	5	6,5	5	8	5	7	3,5	9

O gráfico da Figura 63 indica a qualidade média obtida em cada uma das transições, resultantes dos testes efectuados. A leitura do gráfico deve ser feita tendo em conta que cada uma das animações é representada por uma cor única. Cada uma das barras do gráfico corresponde a um estado a partir do qual é possível atingir o estado pretendido, e a qualidade da transição é representada pelo tamanho dessa barra. No lado esquerdo da figura encontra-se a animação correspondente ao estado atingido após a transição. Por exemplo, durante os testes, foi atingido o estado *bye* partindo da animação *idle*, com o valor médio de qualidade nove, ou seja a transição foi suave. Foi ainda possível atingir o estado *bye* a partir das animações *walk* e *run*, ambas com o valor médio de qualidade cinco, ou seja durante a transição verificou-se o deslizamento de um dos pés do personagem.

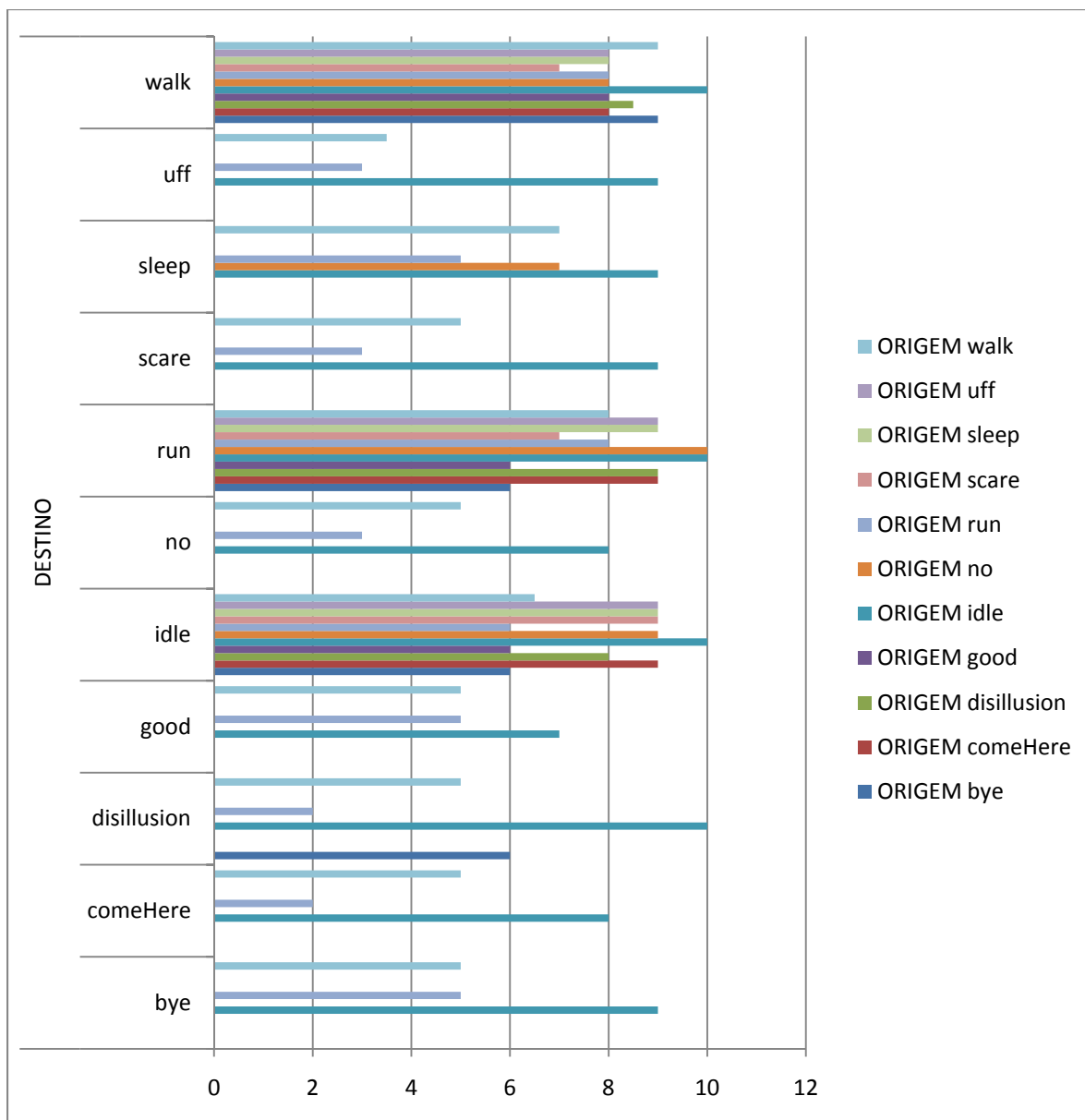


Figura 63 - Resultados das transições entre animações

O gráfico da Figura 64 representa uma porção da informação contida no gráfico da Figura 63. Neste caso são representados apenas os estados iniciais a partir dos quais as transições são mais comuns, ou seja, os estados *walk*, *run* e *idle*. Ao analisar os resultados, verifica-se que, partindo da animação que representa o estado *idle*, é possível obter transições suaves para todos os outros estados (apenas a transição entre *idle* e *good* obteve uma classificação inferior a oito). Tal acontece pois as poses da animação *idle* são propositadamente semelhantes às poses das frames iniciais das restantes animações.

Verifica-se também a que a transição entre o estado *walk* e o estado de *run* se efectua com suavidade (classificação oito). Contudo a transição entre estados que implicam a locomoção (*walk*, *run*), para outros em que o personagem executa o movimento no mesmo local já não obtêm resultados tão

satisfatórios, verificando-se frequentemente o arrastamento de pelo menos um dos pés do personagem. Isto acontece devido à transição abrupta entre a velocidade de *caminhada* ou de *corrida* para a velocidade zero. Devido a este efeito indesejável no caso das transições com origem no estado *walk* a classificação mais frequente é cinco, verificando-se contudo um caso com classificação 7 e outro com classificação 8. O caso mais grave verificado ocorreu na transição entre o estado *walk* e o estado *uff* em que se verificou o deslizamento de ambos os pés (classificação média três aproximadamente).

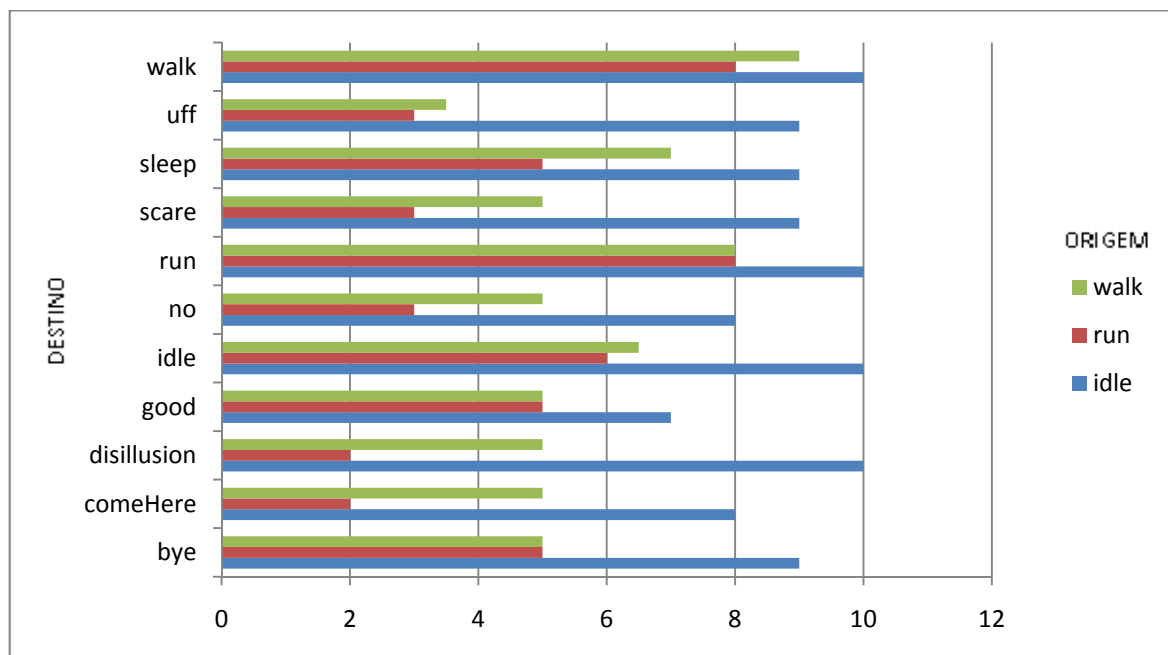


Figura 64 – Transições com origem nas animações *walk*, *run* e *idle*

Relativamente às transições com origem no estado *run* os resultados são um pouco piores que os obtidos nas transições com origem no estado *walk*, pois todas as transições para estados em que a velocidade é zero obtiveram uma classificação igual ou inferior a seis. Porém no teste representado na Figura 65 a transição entre o estado de *run* e outro em que o personagem executa o movimento no mesmo local (*uff*), obtêm resultados muito mais satisfatórios. Com o objectivo de a ajustar progressivamente a velocidade das animações efectuou-se uma transição intermédia entre os estados *run* e *idle*, que obteve a classificação nove. Esta transição intermédia permitiu posteriormente a transição suave entre os estados *idle* e *uff* (classificação nove).

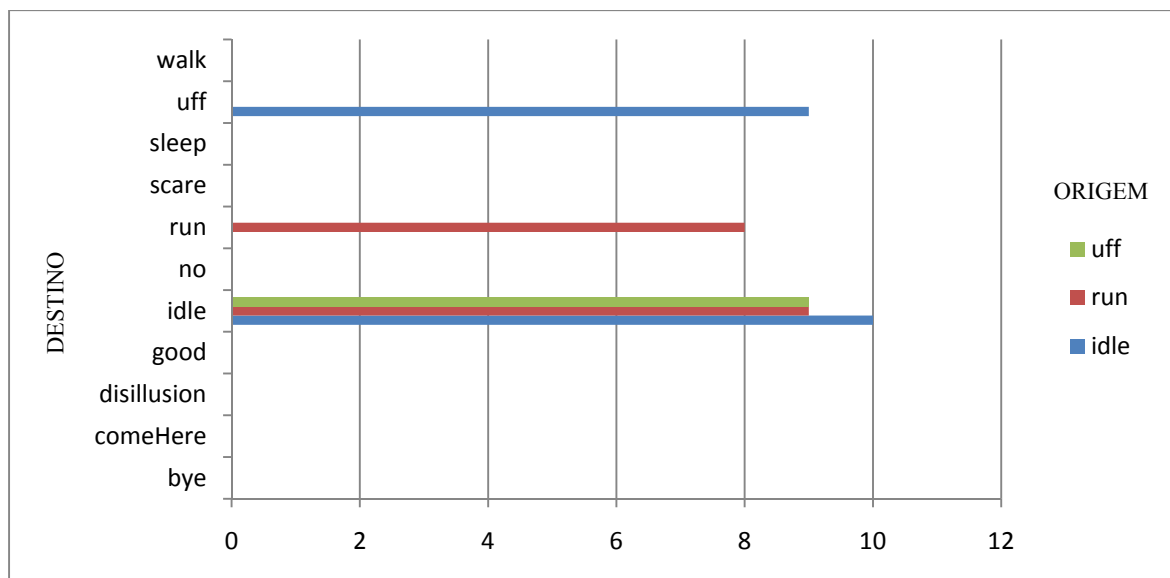


Figura 65 - Transição entre o estado *corrida* e *descanso*

O gráfico da Figura 66 representa as transições com destino nas animações *idle*, *run* e *walk*. As transições em que a origem é um estado em que o personagem executa o movimento no mesmo local, e o destino é o estado *walk*, efectuam-se com suavidade (apenas a transição entre *scare* e *walk* obteve uma classificação inferior a oito). As transições que têm como destino o estado *run* efectuam-se geralmente com suavidade (classificação igual ou superior a oito), com as excepções das transições tendo como origem os estados *bye*, *good* e *scare* em que se verifica a ocorrência de poses intermédias irrealistas (classificação seis e sete).

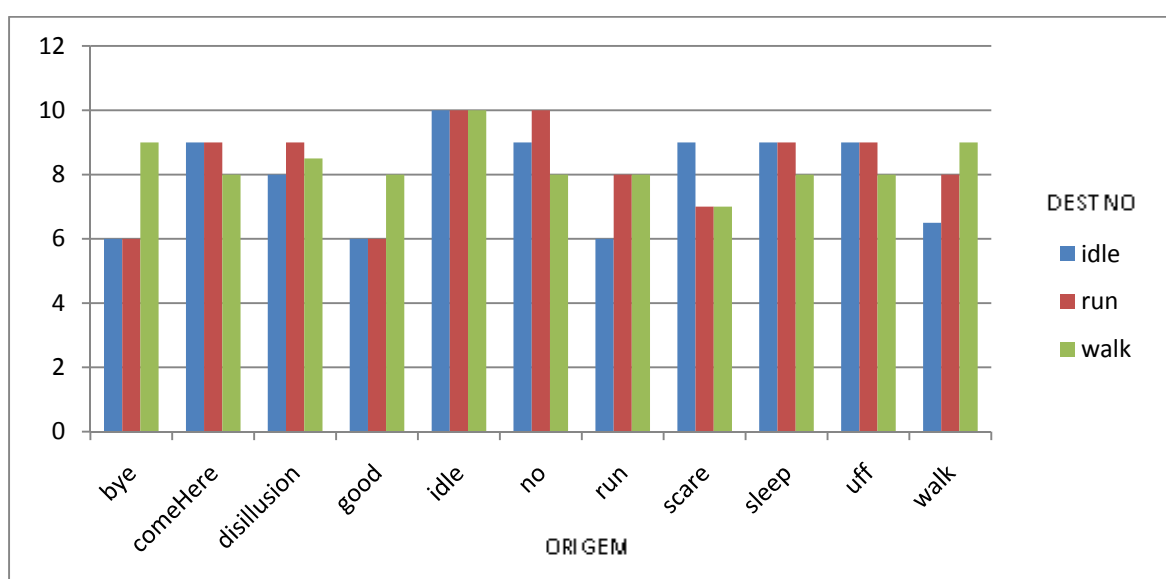


Figura 66 - Transições com destino nas animações *walk*, *run*, *idle*

A análise de resultados do gráfico da Figura 66 permite ainda verificar que as transições com destino no estado *idle* efectuem-se geralmente com suavidade (classificação igual ou superior a oito), com as excepções das transições tendo como origem os estados *bye*, *good*, *run* e *walk* em que se verifica a ocorrência de poses intermédias irrealistas (classificação média seis aproximadamente).

5.1 SUMÁRIO

No presente capítulo foi apresentada uma aplicação prática do sistema de animação de personagens proposto e que consistem na simulação de filmes de animação, utilizando o software *VirtualActor*. Estas simulações permitiram testar várias situações específicas do movimento de personagens virtuais. Embora o foco dos testes efectuados tenha sido um aspecto crítico de um sistema de animação de personagens que consiste na determinação da qualidade das transições, entre animações. Pode-se concluir que as transições entre estados em que o personagem efectua a performance do movimento no mesmo local têm um aspecto agradável. Contudo, as transições entre animações em que a performance do movimento varia de posição ao longo do tempo obtêm resultados menos satisfatórios, pois verifica-se ocasionalmente o efeito *Footskate* (ver secção 2.9). O teste ilustrado na Figura 65, sugere que, para minimizar este efeito indesejado, deve adicionar-se uma transição intermédia para alcançar o estado de *descanso*, a partir do qual é possível atingir todos os outros estados de forma suave.

6 CONCLUSÕES

Nesta dissertação foi apresentado o protótipo de um sistema de animação de personagens virtuais que permite a simulação de movimentos humanos realistas em tempo real, bem como a locomoção no espaço tridimensional. Os movimentos humanos sintetizados permitem fortalecer a capacidade de comunicação não verbal dos personagens virtuais, uma vez que os gestos facilitam a expressão das sensações e estados de espírito experimentados por estes.

A captura óptica de movimentos permitiu construir animações que representam os movimentos primários de um actor humano, evitando um longo processo de animação através de frames chave. Com base nesta técnica foram registados, durante várias sessões, diversos movimentos relevantes da performance de um actor real, que uma vez aplicados a um modelo de esqueleto apropriado, permitiram animar o personagem virtual com elevado realismo. Contudo, o processo de tratamento da informação MoCap revelou-se bastante extenso e aborrecido, devido a problemas relacionados com a oclusão de marcadores.

Os movimentos secundários tais como expressões faciais e fonemas foram adicionados recorrendo à técnica *BlendShapes* disponibilizada pelo software de modelação *Autodesk Maya*. A animação de fonemas permitiu a construção de diálogos entre personagens, capacidade que foi conjugada com a animação de gestos e expressões faciais, possibilitando a representação das emoções e comportamentos dos personagens virtuais.

O controlo da locomoção dos personagens virtuais, foi obtido através da utilização de quaterniões que se revelaram a forma mais directa e eficaz de representar e calcular rotações 3D. Possibilitando a locomoção no espaço tridimensional num único e suave movimento.

Para garantir o planeamento eficiente das restrições dos movimentos foi criada uma estrutura gráfica do tipo *Move Tree*, capaz de definir o fluxo de movimento do personagem. A informação de movimento é misturada em tempo real através de técnicas de interpolação, possibilitando a transição suave entre animações. O sistema possibilita ainda a transição eficaz entre movimentos com diferentes velocidades de locomoção, minimizando o efeito de arrastamento de pés conhecido como *footskate*.

O sistema de animação proposto, uma vez integrado no software *VirtualActor*, permite a qualquer utilizador a sintetização de movimentos humanos através da introdução de instruções simples. Este sistema poderá também ser adaptado para uma utilização em jogos de computador, em filmes, na educação, em histórias interactivas, e também na terapia de fobias sociais e no treino comportamental, como por exemplo, para conduzir certos tipos de diálogos.

Cada utilizador poderá criar filmes de animação em 3D com actores virtuais emocionalmente expressivos, ou seja, poderão transmitir ilusão de vida, expressar-se emocionalmente, com voz e com sinais não verbais. Esta tecnologia apresenta-se vantajosa, visto não ser necessário despende longos períodos de formação nem de muito investimento.

6.1 MODIFICAÇÕES SUGERIDAS/ TRABALHO FUTURO

Existem várias ideias que podem ser desenvolvidas em termos de trabalho futuro. Apesar de não ter feito parte dos objectivos da dissertação, a implementação de um *Motion Graph* poderia melhorar o sistema proposto.

O grafo de selecção de transições utilizado é do tipo *move tree*, tendo este grafo sido construído manualmente, pois o foco desta dissertação foi o planeamento eficiente das restrições dos movimentos e não propriamente a construção automática do grafo. Contudo, o desenvolvimento de boas *move trees* envolve um processo meticoloso e o investimento de muito tempo. Os *Motion Graphs* poderão ser utilizados para automatizar este processo e para lhe conferir mais flexibilidade a altura de acrescentar novas animações. Contudo, a utilização de grafos de movimento é potencialmente problemática de várias formas. Se forem identificadas transições agradáveis através da comparação de cada frame com todas as outras frames, pode obter-se um grafo de movimento extremamente denso. Isto pode acontecer sempre que dois movimentos similares ocorram no mesmo clipe de movimento. Muitas das ligações do grafo podem ser removidas sem qualquer perda significativa de funcionalidade. Uma

escolha passa por não comparar uma frame com a sua frame seguinte. Este problema pode também ocorrer quando existem secções com informação muito similar. Por exemplo, se existir uma porção da informação de movimento com um personagem a efectuar um movimento muito idêntico, isto pode implicar um grande aumento do número de conexões entre todas as frames similares.

Está ainda prevista a utilização do sistema proposto para controlar o movimento de personagens virtuais, numa aplicação com recurso a Realidade Aumentada, cujo objectivo consiste em apoiar humanos no desempenho de tarefas relacionadas com a produção de produtos têxteis.

Referências Documentais

- [Agarwala et al., 2004] Agarwala, Aseem; Hertzmann, Aaron; Salesin, David H.; Seitz, Steven M. - Keyframe-Based Tracking for Rotoscoping and Animation
- [Arikan et al., 2003] Arikan, Okan; Forsyth, David A.; O'Brien, James F. - Motion Synthesis from Annotations, 2003
- [Camilo and Conci, 2006] Camilo, Márcio S.; Conci, Aura - Guff's Character Animation System for 3D Games and Applications, 2006
- [CEGUI] Crazy Eddie's GUI System. 17/10/2010.
<http://www.ogre3d.org/tikiwiki/-CEGUI>
- [Chilton et al., 2004] Chilton, Nicholas; Wan, Tao; Earnshaw, Rae - Real Time Visualization and Blending of Motion Capture Animations, 2004
- [cgCharacter] The Ultimate Human. 08/08/2010.
<http://www.cgcharacter.com>
- [Dam, 1998] Dam, Erik B.; Koch, Martin; Lillholm, Martin - Quaternions, Interpolation and Animation, 1998
- [Fleisher, 1917] Fleisher, Max - Method of Producing Moving Picture Cartoons. US Patent no. 1,242,674, 1917
- [Giese et al., 2009] Giese, Martin A.; Mukovskiy, Albert; Park, Aee-Ni - Real-Time Synthesis of Body Movements Based on Learned Primitives, 2009
- [Hugill et al., 2010] Hugill, Nadine; Fink, Bernhard and Neave, Nick - The Role of Human Body Movements in Mate Selection.
- [Iurgel et al., 2010] Iurgel, Ido A.; Silva, Rogério E.; Santos, Manuel F. - Towards Virtual Actors for Acting out Stories, 2010

[Junker, 2006]	Junker, Gregory - Pro OGRE 3D Programming, 2006
[Kerlow, 2004]	Kerlow, Isaac Victor - The Art of 3D Computer Animation and Effects (third edition), 2004
[Kovar et al., 2002A]	Kovar, Lucas; Gleicher, Michael; Pighin, Frédéric - Motion Graphs, 2002
[Kovar et al., 2002B]	Kovar, Lucas; Gleicher, Michael; Schreiner, John - Footskate Cleanup for Motion Capture Editing, 2002
[Kovar and Gleicher, 2003]	Kovar, Lucas; Gleicher, Michael - Flexible Automatic Motion Blending with Registration Curves, 2003
[Kovar and Gleicher, 2004]	Kovar, Lucas; Gleicher, Michael - Automated Extraction and Parameterization of Motions in Large Data Sets, 2004
[Kremer, 2008]	Kremer, Verena Elisabeth - Quaternions and SLERP, 2008
[Kuipers, 2002]	Kuipers, Jack B. - Quaternions and Rotation Sequences : A Primer with Applications to Orbits, Aerospace and Virtual Reality, 2002
[Lasseter, 1987]	Lasseter, John - Principles of Traditional Animation Applied to 3D Computer Animation, 1987
[Lau and Kuffner, 2005]	Lau, Manfred; Kuffner, James J. - Behavior Planning for Character Animation, 2005
[Lever, 2002]	Lever, Nik - Real-time 3D Character Animation with Visual C++, 2002
[Maestri, 1999]	Maestri, Geoge - Digital Character Animation 2, Volume 1, 1999
[Maya]	Autodesk Maya. 17/10/2010. http://usa.autodesk.com/adsk/servlet/pc/index?id=13577897&siteID=123112

[MotionBuilder]	Autodesk MotionBuilder. 17/10/2010. http://usa.autodesk.com/adsk/servlet/pc/index?id=13581855&siteID=123112
[NewtonGD]	Newton Game Dynamics – Physics Engine. 17/10/2010. http://newtondynamics.com/forum/newton.php
[O’Brien, 2000]	O’Brien, James F. -Automatic Joint Parameter Estimation from Magnetic Motion Capture Data”, 2000
[O'Rourke, 1998]	O'Rourke, Michael - Principles of Three-Dimensional Computer Animation: Modeling, Rendering, and Animating with 3D Computer Graphics, 1998
[Parent et al., 2009]	Parent, Rick; Ebert, David S.; Gould, David A. D. - Computer Animation Complete, 2009
[Petkovsek, 2005]	Petkovsek, Cory - Quaternion and Rotation Primer. 08/08/2010. http://www.ogre3d.org/tikiwiki/Quaternion+and+Rotation+Primer
[QuickGUI]	QuickGUI. 17/10/2010. http://www.ogre3d.org/tikiwiki/QuickGUI&structure=Libraries
[Reynolds, 1999]	Reynolds, Craig W. - Steering Behaviors for Autonomous Characters, 1999
[Safonova and Hodgins, 2005]	Safonova, Alla; Hodgins, Jessica K. - Analyzing the Physical Correctness of Interpolated Human Motion, 2005
[Shoemake, 1985]	Shoemake, Ken - Animating Rotation with Quaternion Curves, 1985
[Torre et al., 2009]	Torre, Fernando; Hodgins, Jessica K.; Montano, Javier - Guide to the Carnegie Mellon University Multimodal Activity (CMU-MMAC) Database, 2009

- [Vicon iQ, 2004] Vicon Motion Systems Inc - Vicon iQ Motion Capture Production and Control Software, 2004
- [Vicon iQ 2.5, 2004] Vicon Motion Systems Inc - Vicon iQ 2.5 Motion Capture Production and Control Software, 2004
- [VirtualActor, 2008] Virtual Actor. 08/08/2010.
http://www.ccg.pt/index.php?option=com_content&view=article&id=120%3Avirtual-actor&catid=26&Itemid=3&lang=p
- [WallaceAndGromit] Wallace and Gromit. 17/10/2010.
<http://www.wallaceandgromit.com/>
- [Wang and Bodenheimer, 2004] Wang, Jing; Bodenheimer, Bobby - An Evaluation of a Cost Metric for Selecting Transitions between Motion Segments, 2004
- [Welbergen et al., 2009] Welbergen, H. van; Basten, B. J. H. van; Egges, A. - Real Time Animation of Virtual Humans: A Trade-off Between Naturalness and Control, 2009
- [Whitaker and Halas, 2002] Whitaker, Harold; Halas, John - Timing for Animation, 2002
- [Whitaker and Halas, 2009] Whitaker, Harold; Halas, John - Timing for Animation Second Edition, 2009

ANEXO A

Aquisição de Movimentos em *Vicon iQ*

- **ETAPA 0 - Iniciar Sistema**

1. Ligar o sistema de câmaras no dispositivo central MX Box, ilustrado na Figura 1;



Figura Anexo - 1 - Vicon Mx Box

2. Iniciar a aplicação *Vicon iQ* e localizar o separador *Setup*. Para ligar o sistema clicar no botão **Connect/Disconnect** representado na Figura 3.

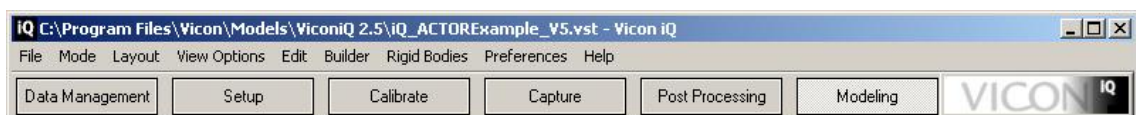


Figura Anexo - 2 - Interface Vicon iQ 2.5



Figura Anexo - 3 - Arranque do sistema

3. No separador *Data Management*, criar um novo *Set* clicando no ícone representado por um círculo verde. De seguida clicar com o botão direito do rato sobre este de forma a criar um novo *Patient* e uma nova *Session*.

- **ETAPA 1 - Preparação do cenário**

1. Definir o espaço físico onde serão realizadas as capturas conforme a Figura 5, através da distribuição de *markers* no chão. Na figura 4 é possível visualizar um *marker* (ou marcador).



Figura Anexo - 4 - Marcador



Figura Anexo - 5 - Disposição do cenário

2. No que diz respeito às câmaras (Figura 6) a luz verde significa que a câmara está a capturar correctamente, a luz Azul significa que esta câmara está a ser configurada e por fim, a luz Amarela está associada a um possível erro ou problema.

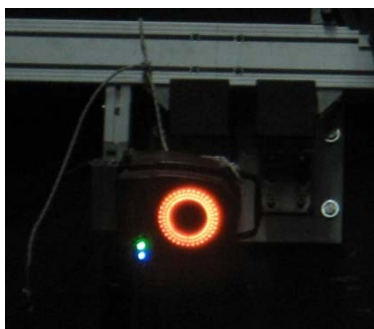


Figura Anexo - 6 - Câmara Vicon MX

3. De seguida deve proceder-se ao ajuste cada uma das câmaras no separador *Setup*, os parâmetros são os seguintes:
 - ***Strobe Intensity*** (Intensidade da luz infra-vermelha, este parâmetro deve ser colocado ao máximo);
 - ***Camera Threshold***;
 - ***Gain***;
 - ***Circle Quality***;
4. Deve ser ajustada directamente nas câmaras, conforme a configuração espacial desejada, a orientação, a abertura do diafragma e a profundidade. Em seguida deve utilizar-se uma varinha com marcadores para facilitar delimitação da região de captura. Após este processo o resultado obtido deve ser semelhante ao ilustrado na Figura 7. A varinha deve aparecer dentro do quadrado branco, deve verificar-se se também se as regiões junto ao piso e ao nível da

cabeça do sujeito, estão a ser capturadas correctamente. No final da verificação de todas as câmaras devem ser retirados do chão os marcadores, que ajudaram a definir a área de captura.

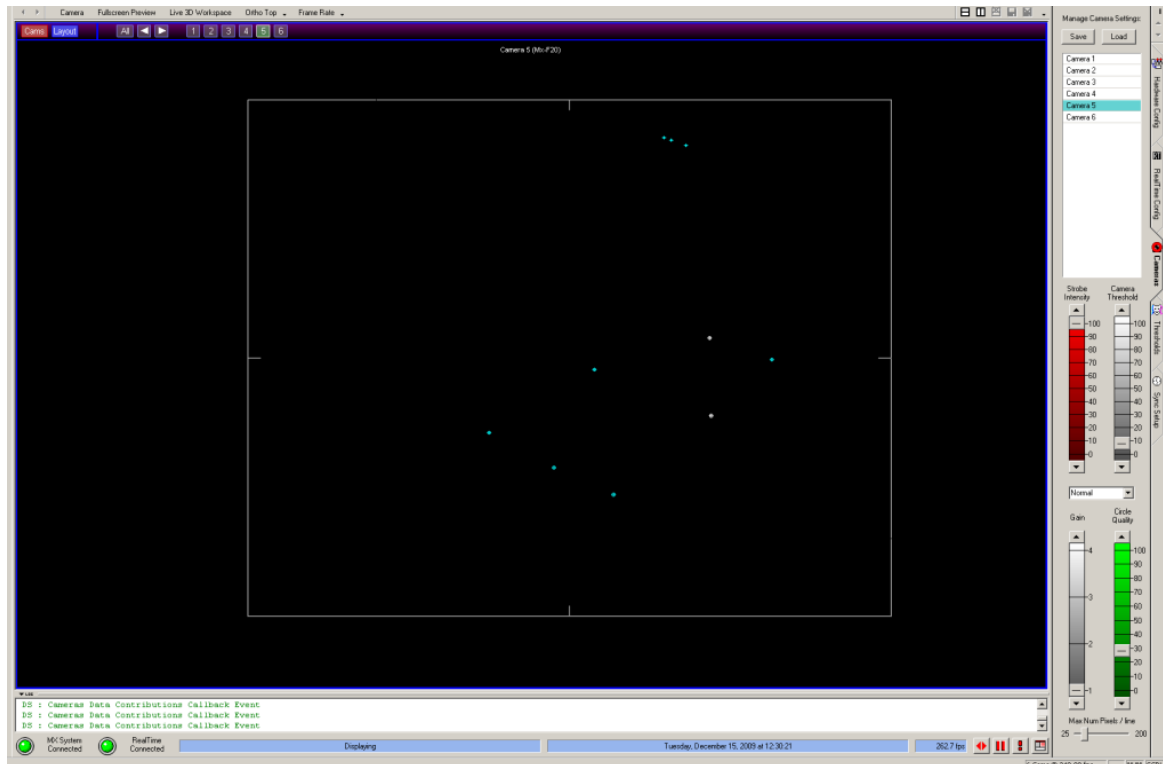


Figura Anexo - 7 - Definição da região de captura

• ETAPA 2 - Calibração

1. Neste passo deve seleccionar-se o separador *Calibrate*.
- De seguida será necessário escolher a varinha que vai ser utilizada para introduzir a informação espacial no sistema. A dimensão da varinha é descriminada na *comboBox Wand*, conforme se pode visualizar na Figura 8.
- O passo seguinte é **Calibrate Cameras > Start Wand wave**.

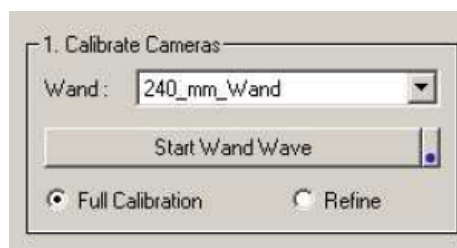


Figura Anexo - 8 – Arranque da calibração das câmaras

- Deve agitar-se a varinha em movimentos suaves sobre a região definida anteriormente. Quanto maior for a amplitude e duração dos movimentos mais perfeita será a calibração. Devem ser captadas no mínimo 10.000 *wands*. Na Figura 9 ilustra-se o ambiente que deve ser obtido.

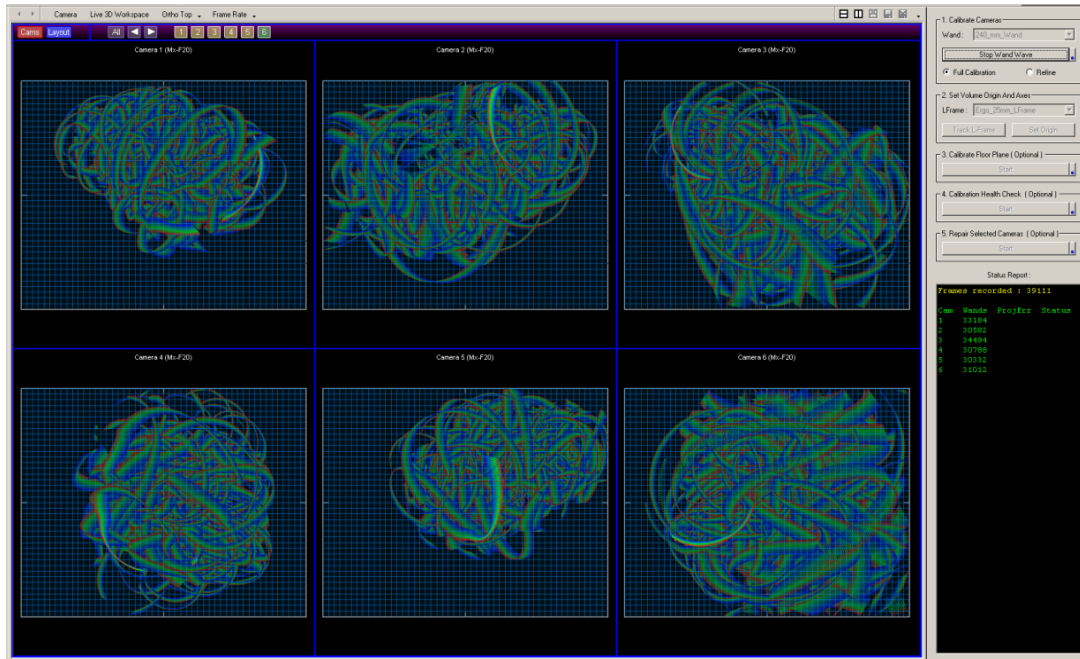


Figura Anexo - 9 - Calibração das câmaras

- Um dos parâmetros fundamentais na calibração é a definição do referencial 3D, para isso deve utilizar-se o esquadro apresentado na Figura 10. Este deve ser colocado no cenário conforme a configuração pretendida. Deve ser verificada a visibilidade dos marcadores por todas as câmaras.

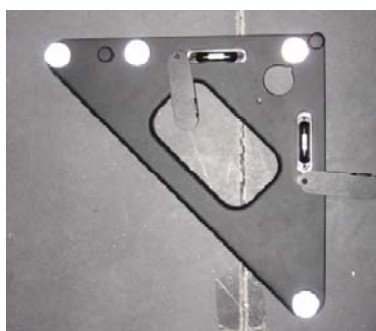


Figura Anexo - 10 - Esquadro com marcadores

- Clicar na opção **Set Volume Origin and Axes > Track L-Frame** (Figura 11), que vai activar um processo de calibração do piso. Após cerca de dez segundos clicar em **Set Origin**.



Figura Anexo - 11 - Calibração do chão

3. Como se pode verificar na Figura 12, após o processo de calibração todas as câmaras devem reconhecer a sua localização e orientação exacta no cenário de captura.

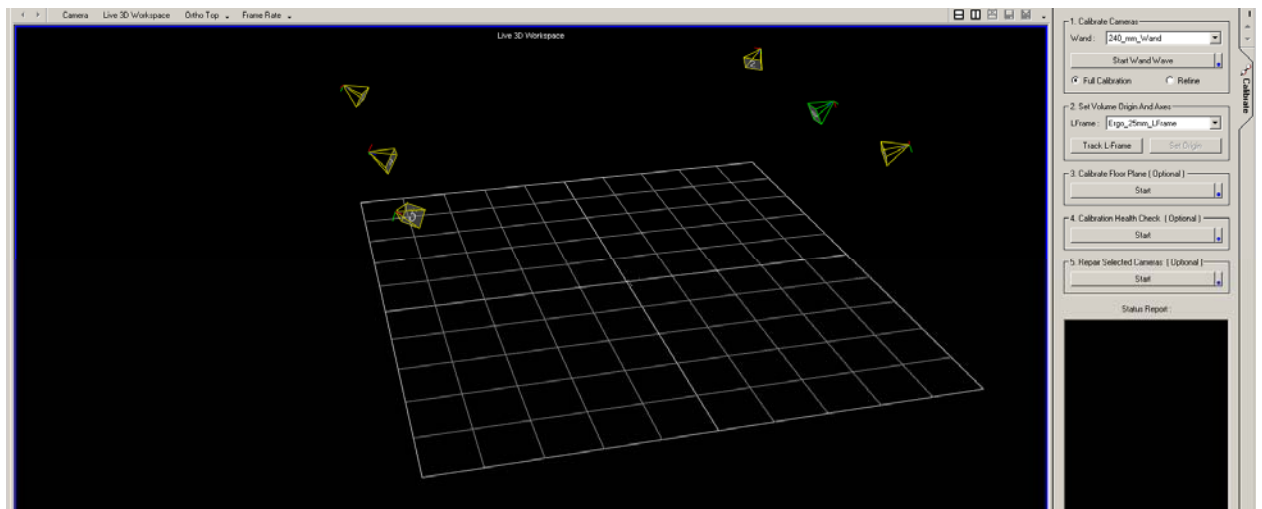


Figura Anexo - 12 - Resultado final após a calibração

• ETAPA 3 - Captura

1. Preparação do Actor.
 - Os marcadores devem estar dispostos de acordo com o standard utilizado pelo *Vicon iQ*, totalizando assim um conjunto de 41 marcadores.



Figura Anexo - 13 - Pormenor do pé e mão

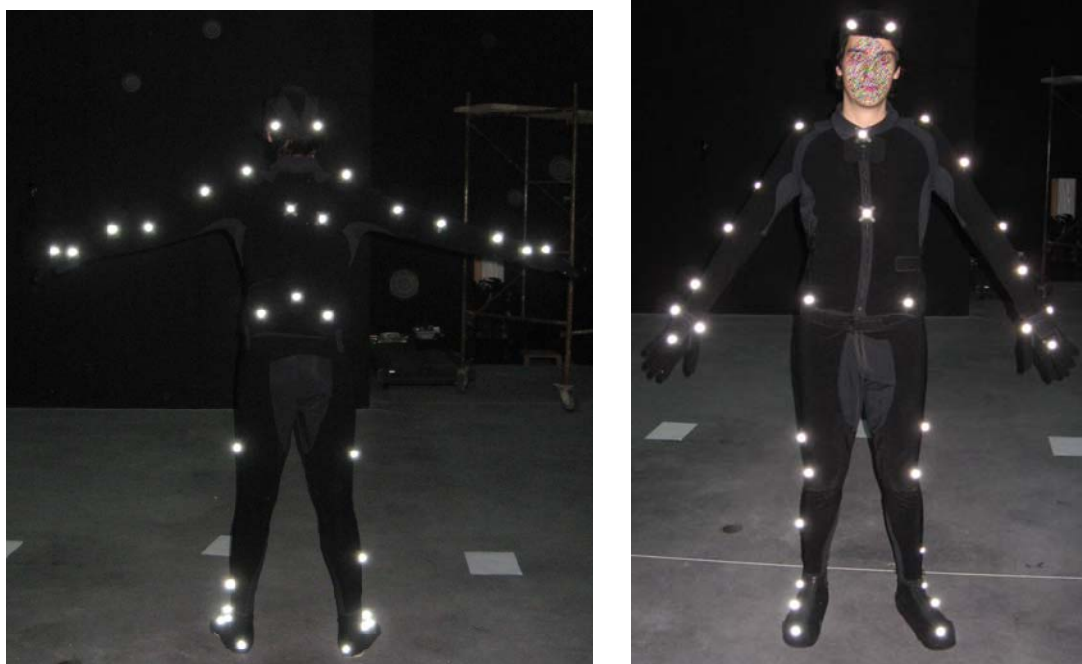


Figura Anexo - 14 - Disposição dos marcadores, vista frontal e vista de retaguarda.

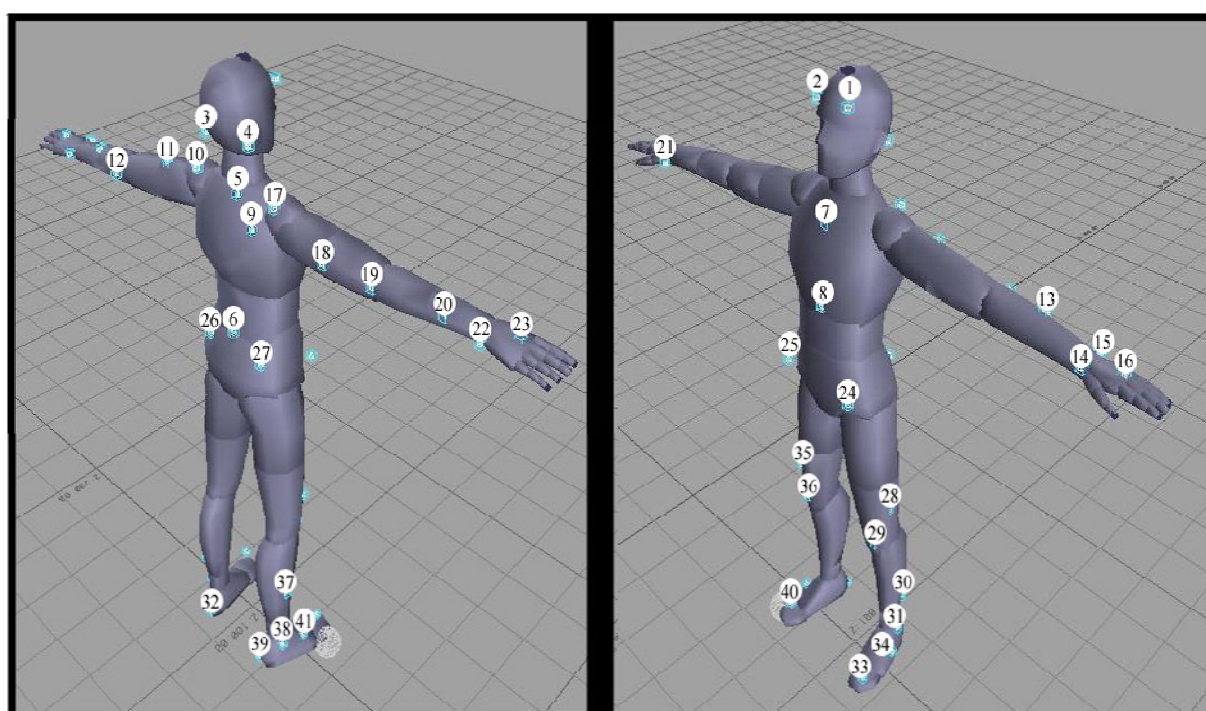


Figura Anexo - 15 - Distribuição de Marcadores

Numero	Rotulo	Localização do Marcador
1	LFHD	Cabeça – À frente do lado esquerdo
2	RFHD	Cabeça – À frente do lado direito
3	LBHD	Cabeça – Atrás do lado esquerdo
4	RBHD	Cabeça – Atrás do lado direito
5	C7	Sétima Cervical
6	T10	Decima Torácica
7	CLAV	União das Clavículas
8	STRN	Terminação inferior do Esterno
9	RBAC	Parte posterior do tronco do lado direito
10	LSHO	Ombro esquerdo
11	LUPA	Meio do braço esquerdo
12	LELB	Cotovelo esquerdo
13	LFRM	Antebraço esquerdo
14	LWRA	Mão – rádio – lado esquerdo
15	LWRB	Mão – cúbito – lado esquerdo
16	LFIN	Falange do indicador esquerdo
17	RSHO	Ombro direito
18	RUPA	Meio do braço direito
19	RELB	Cotovelo direito
20	RFRM	Antebraço direito
21	RWRA	Mão – rádio – lado direito
22	RWRB	Mão – cúbito – lado direito
23	RFIN	Falange do indicador direito
24	LFWT	Bacia – À frente do lado esquerdo
25	RFWT	Bacia – À frente do lado direito
26	LBWT	Bacia – Atrás do lado esquerdo
27	RBWT	Bacia – Atrás do lado direito
28	LTHI	Meio da coxa esquerda
29	LKNE	Joelho esquerdo
30	LSHN	Canela esquerda
31	LANK	Tornozelo esquerdo
32	LHEE	Calcanhar esquerdo
33	LTOE	Dedo do pé esquerdo
34	LMT5	Quinto metatarso esquerdo
35	RTHI	Meio da coxa direita
36	RKNE	Joelho direito
37	RSHN	Canela direita
38	RANK	Tornozelo direito
39	RHEE	Calcanhar direito
40	RTOE	Dedo do pé direito
41	RMT5	Quinto metatarso direito

Figura Anexo - 16 - Localização dos marcadores

2. Captura de Movimentos

- No modo *Capture* deve verificar-se a visibilidade de todos os marcadores colocados sobre o fato especial que o actor veste.
- Na Primeira captura o actor deve fazer a performance *Tpose*, devem ser capturadas cerca de 500 frames.
- Embora seja opcional o processo seguinte deve ser executado para facilitar a edição posterior.
- Criação do sujeito para *auto labeling*:
 - No separador *Data Management*, abrir o ficheiro com a performance *Tpose* carregar no ícone que tem uma cruz azul;
 - Depois correr o ficheiro conforme a Figura 17

Post Processing > Reconstruct > Run

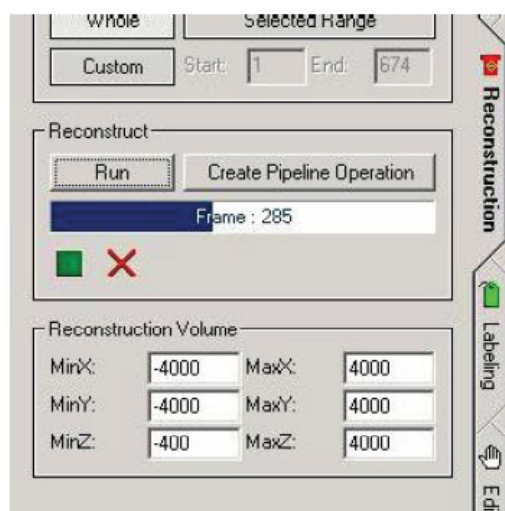


Figura Anexo - 17 - Reconstrução da informação

- Criação do sujeito

Post processing > Subject > Create New

- *Labelling* do sujeito (activar o modo *sequence*)

Post processing > Labeling

- Calibração do sujeito
- Conforme a Figura 18, em **Post processing > Events** activar botão **G**



Figura Anexo - 18 - Calibração do sujeito

- Activar o *Tpose event*
- Calibração do sujeito, em **Post processing > Subject > Calibrate Subject**
- Exportação do sujeito
 - **Post processing > Subject > Export**
- Voltando ao separador *Capture* deve ser escolhida a opção *Active Object* e ao seleccionar o sujeito criado anteriormente se o actor voltar a executar a “pose em T” o esqueleto aparece devidamente rotulado.
- Neste momento esta concluída a configuração do sistema, é assim possível capturar novos movimentos

• ETAPA 4 - Tratamento da informação capturada

1. No separador *Data Management* escolher a captura efectuada na sessão, clicando sobre o ícone azul em forma de cruz.
 2. Escolher o separador *Post Processing*
- Depois correr o ficheiro conforme a Figura 17
 - **Post Processing > Reconstruct > Run**
 - Correr a *pipeline* de procedimentos predefinida com uma serie de tarefas nomeadamente:
 - *Label trajectories*, uma vez associado um *labeling* a um sujeito esta tarefa permite fazer o *labeling* automático de qualquer outra captura feita pelo actor associado ao sujeito.
 - Os rótulos que não ficarem associados deverão ser associados manualmente
 - *Fill gap*, é uma tarefa que permite corrigir os instantes da captura, em que determinados marcadores foram ocluídos, a correcção passa pela predição da posição de um determinado marcador através interpolação (*using splines; rigid body; using Splines*).
 - *Clean ghosts*, esta tarefa permite eliminar capturas de pontos que não fazem parte do sistema de pontos que pretendemos capturar (*Delete unlabeled trajectories*).
 - Exportação para o formato C3D.